invensys

Eurotherm

nanodac™

# User Guide

nanodac™ recorder/controller
Versions 4.10 and later

HA030554/6
June 2012

invensys
Eurotherm

# Declaration of Conformity

| | |
|---|---|
| Manufacturer's name: | Eurotherm Limited |
| Manufacturer's address: | Faraday Close, Worthing, West Sussex, BN13 3PL, United Kingdom |
| Product type: | Recorder / controller |
| Models: | nanodac Status level A1 and above |
| Safety specification: | EN61010-1: 2001 |
| EMC emissions specification: | EN61326-1: 2006 Class B (100 to 230V ac supply)<br>EN61326-1: 2006 Class A (24V ac/dc supply) |
| EMC immunity specification: | EN61326-1: 2006 Industrial locations |

Eurotherm Limited hereby declares that the above products conform to the safety and EMC specifications listed. Eurotherm Limited further declares that the above products comply with the EMC Directive 2004/108/EC, and also with the Low Voltage Directive 2006/95/EC.

Signed: K Shaw Dated: 11/11/10

Signed for and on behalf of Eurotherm Limited.
Kevin Shaw
(R&D Director)

CE

# Restriction of Hazardous Substances (RoHS)

**Product group** nanodac

**Table listing restricted substances**

Chinese

限制使用材料一览表

| 产品 | 有毒有害物质或元素 | | | | | |
|---|---|---|---|---|---|---|
| nanodac | 铅 | 汞 | 镉 | 六价铬 | 多溴联苯 | 多溴二苯醚 |
| 印刷线路板组件 | X | O | O | O | O | O |
| 附属物 | O | O | O | X | O | O |
| 显示器 | O | O | O | O | O | O |
| O | 表示该有毒有害物质在该部件所有均质材料中的含量均在SJ/T11363-2006标准规定的限量要求以下。 | | | | | |
| X | 表示该有毒有害物质至少在该部件的某一均质材料中的含量超出SJ/T11363-2006标准规定的限量要求。 | | | | | |

English

Restricted Materials Table

| Product | Toxic and hazardous substances and elements | | | | | |
|---|---|---|---|---|---|---|
| nanodac | Pb | Hg | Cd | Cr(VI) | PBB | PBDE |
| PCBA | X | O | O | O | O | O |
| Enclosure | O | O | O | X | O | O |
| Display | O | O | O | O | O | O |
| O | Indicates that this toxic or hazardous substance contained in all of the homogeneous materials for this part is below the limit requirement in SJ/T11363-2006. | | | | | |
| X | Indicates that this toxic or hazardous substance contained in at least one of the homogeneous materials used for this part is above the limit requirement in SJ/T11363-2006. | | | | | |

Approval

| Name: | Position: | Signature: | Date: |
|---|---|---|---|
| Martin Greenhalgh | Quality Manager | Martin Greenhalgh | 11th APRIL 2010 |

IA029470U790/1 (CN26215)

# nanodac Recorder/Controller

## User Guide

## List of sections



## Associated documents

HA028838 Printable version of iTools Help
HA025464 EMC installation guidelines
HA027962 Printable version of ‘Review’ Help

## Application notes

HA030817U001 Archiving data from the nanodac recorder/controller
HA030817U002 Heat/Cool with carbon potential or oxygen level monitoring
HA030817U003 Heat only temperature control and carbon potential control
HA030817U004 Virtual channels using the nanodac recorder/controller.

## Software effectivity

This manual refers to instruments fitted with software version 3.0.
Software versions 2.20 onwards are ‘backwards compatible’ so that it can be used on all hardware versions of the unit.
Previous software versions are not compatible with instruments with hardware status greater than 2.
The status level may be found on the instrument label and consists of a letter indicating software status followed by a numeral indicating the hardware status (e.g. ‘B2’)

# nanodac Recorder/Controller

# User Guide

# Contents List

## List of Contents (Cont.)

# List of Contents (Cont.)

# List of Contents (Cont.)

## List of Contents (Cont.)

Section Page

# List of Contents (Cont.)

# List of Contents (Cont.)

# List of Contents (Cont.)

This page is deliberately left blank

# SAFETY NOTES

**WARNINGS**

1. Any interruption of the protective conductor inside or outside the apparatus, or disconnection of the protective earth terminal is likely to make the apparatus dangerous under some fault conditions. Intentional interruption is prohibited.
2. Live sensors: The unit is designed to operate if the temperature sensor is connected directly to an electrical heating element. It must be ensured that service personnel do not touch connections to such inputs whilst the inputs are live. With live sensors, all cables, connections and switches for connecting the sensor must be mains rated for use in 240V Cat II.
3. Grounding the temperature sensor shield: Where it is common practice to replace the temperature sensor whilst the instrument is live, it is recommended that the shield of the temperature sensor be grounded to safety earth, as an additional protection against electric shock.
4. The instrument must not be wired to a three-phase supply with an unearthed star connection, because, under fault conditions, such a supply could rise above 240V RMS with respect to ground, thus rendering the instrument unsafe.

Notes:

1. Safety requirements for permanently connected equipment state:
    a. A switch or circuit breaker shall be included in the building installation.
    b. It shall be in close proximity to the equipment and within easy reach of the operator.
    c. It shall be marked as the disconnecting device for the equipment.
2. Recommended external fuse ratings are: 2A Type T 250V.

1. This instrument is intended for industrial temperature and process control applications within the requirements of the European directives on safety and EMC.
2. Installation may be carried out only by qualified personnel.
3. To prevent hands or metal tools coming into contact with parts that are electrically live the instrument must be installed in an enclosure.
4. Where conductive pollution (e.g. condensation, carbon dust) is likely, adequate air conditioning/filtering/sealing etc. must be installed in the enclosure.
5. The mains supply fuse within the power supply is not replaceable. If it is suspected that the fuse is faulty, the manufacturer's local service centre should be contacted for advice.
6. Whenever it is likely that protection has been impaired, the unit shall be made inoperative, and secured against accidental operation. The manufacturer's nearest service centre should be contacted for advice.
7. If the equipment is used in a manner not specified by the manufacturer, the protection provided by the equipment might be impaired.
8. The unit must be wired according to the instructions in this manual.
9. Before any other connection is made, the protective earth terminal shall be connected to a protective conductor. The mains (supply voltage) wiring must be terminated in such a way that, should it slip, the Earth wire would be the last wire to become disconnected. The protective earth terminal must remain connected (even if the equipment is isolated from the mains supply), if any of the I/O circuits are connected to hazardous voltages*.
   The protective earth connection must always be the first to be connected and the last to be disconnected.
   Wiring must comply with all local wiring regulations, e.g. in the UK, the latest IEEE wiring regulations (BS7671) and in the USA, NEC class 1 wiring methods.
10. Signal and supply voltage wiring should be kept separate from one another. Where this is impractical, shielded cables should be used for the signal wiring.

* A full definition of 'Hazardous' voltages appears under 'Hazardous live' in BS EN61010. Briefly, under normal operating conditions, hazardous voltages are defined as being > 30V RMS (42.2V peak) or > 60V dc.

**SAFETY NOTES (Cont.)**

11. The maximum continuous voltage applied between any of the following terminals must not exceed 240Vac.
    1. Relay output to logic, dc or sensor input connections
    2. Any connection to ground.

    The ac supply must not be connected to sensor input or low-level inputs or outputs.
12. Over temperature protection: A separate over-temperature protection unit (with an independent temperature sensor) should be fitted to isolate the process heating circuit should a fault condition arise.

    Alarm relays within the recorder/controller do not give protection under all fault conditions/
13. In order to allow the power supply capacitors to discharge to a safe voltage, the supply must be disconnected at least two minutes before the instrument is removed from its sleeve. The touching of the exposed electronics of an instrument which has been removed from its sleeve should be avoided.
14. Instrument labels may be cleaned using iso-propyl alcohol, or water or water-based products. A mild soap solution may be used to clean other exterior surfaces.

## USB DEVICE PRECAUTIONS

Note: the use of U3 USB Flash drives is not recommended.

1. Precautions against electrostatic discharge should be taken when the instrument terminals are being accessed. The USB and Ethernet connections are particularly vulnerable.
2. Ideally, the USB device should be plugged directly into the instrument, as the use of extension leads may compromise the instrument's ESD compliance. Where the instrument is being used in an electrically 'noisy' environment however, it is recommended that the user brings the USB socket to the front of the panel using a short extension lead. This is because the USB may 'lock up' or reset in noisy environments and the only means of recovery is to remove the device, then re-insert it. For memory sticks, EMC-related failure during a write operation might cause corruption of the data held on the stick. For this reason, the data on the memory stick should be backed up before insertion and checked after removal.
3. When using a USB extension cable, a high quality screened cable must be used. The total length of USB cable between the device and the USB port must not exceed 3 metres (10 ft.)
4. Most barcode readers and keyboards are not designed for use in industrial EMC environments, and their operation in such environments may result in impaired performance of the recorder/controller.

## 32-BIT RESOLUTION

Floating point values are stored in IEEE 32-bit single precision format. Values which require greater resolution than is available in this format are rounded up or down.

## SYMBOLS USED ON THE RECORDER LABELLING

One or more of the symbols below may appear as a part of the recorder labelling.

| | | | |
|---|---|---|---|
| | Refer to manual for instructions | | Risk of electric shock |
| | This unit is CE approved | | Precautions against static electrical discharge must be taken when handling this unit |
| | C-Tick mark for Australia (ACA) and New Zealand (RSM) | | Ethernet connector |
| c UL US LISTED | Underwriters laboratories listed mark for Canada and the U.S.A. | | USB connector |
| 40 | For environmental reasons, this unit must be recycled before its age exceeds the number of years shown in the circle. | | Protective conductive terminal (Safety Earth) |

# 1 INTRODUCTION

This document describes the installation, operation and configuration of a paperless graphic recorder/controller. The instrument comes, as standard with four input channels and is equipped, for secure archiving via FTP transfer and/or to USB memory stick.

## 1.1 UNPACKING THE INSTRUMENT

The instrument is despatched in a special pack, designed to give adequate protection during transit. Should the outer box show signs of damage, it should be opened immediately, and the contents examined. If there is evidence of damage, the instrument should not be operated and the local representative contacted for instructions. After the instrument has been removed from its packing, the packing should be examined to ensure that all accessories and documentation have been removed. The packing should then be stored against future transport requirements.

# 2 INSTALLATION

**CAUTION**

Before installation, ensure that the specified instrument supply voltage matches the facility supply.

## 2.1 MECHANICAL INSTALLATION

Figure 2.1 gives installation details.

### 2.1.1 Installation procedure

1. If it is not already in place, fit the IP65 sealing gasket behind the front bezel of the instrument.
2. Insert the instrument through the panel cutout, from the front of the panel.
3. Spring the retaining clips into place, and secure the instrument by holding it firmly in place whilst pushing both clips towards the rear face of the panel.
4. The protective membrane can now be removed from the display.

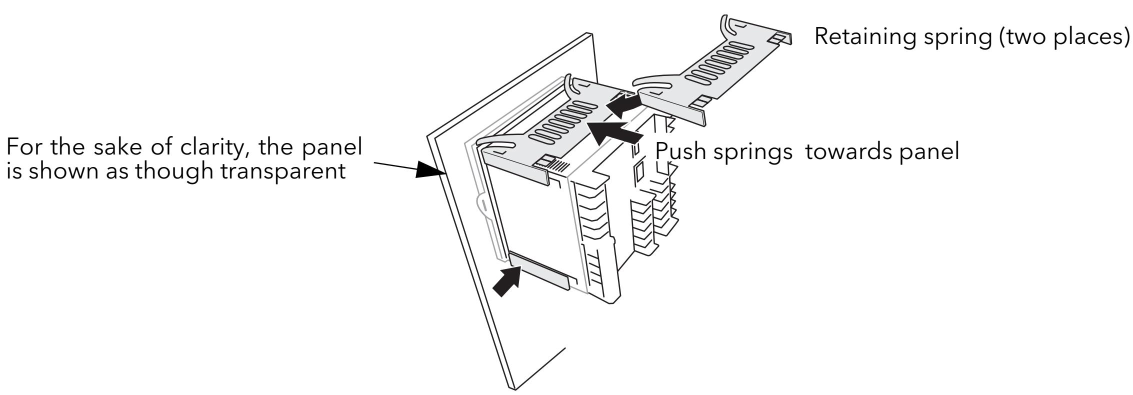



Figure 2.1.1 Securing the instrument

### 2.1.2 Demounting

**WARNING**

Before removing the supply voltage wiring, isolate the supply voltage and secure it against unintended operation.

1. Isolate the mains supply and secure it against accidental operation. Remove all wiring and the USB device and Ethernet cable (if any).
2. Remove the retaining springs by unhooking them from the sides using a small flat-blade screwdriver.
3. Pull the instrument forwards out of the panel.

Note: See section C1 (Battery replacement) for a more detailed description

## 2 MECHANICAL INSTALLATION (Cont.)

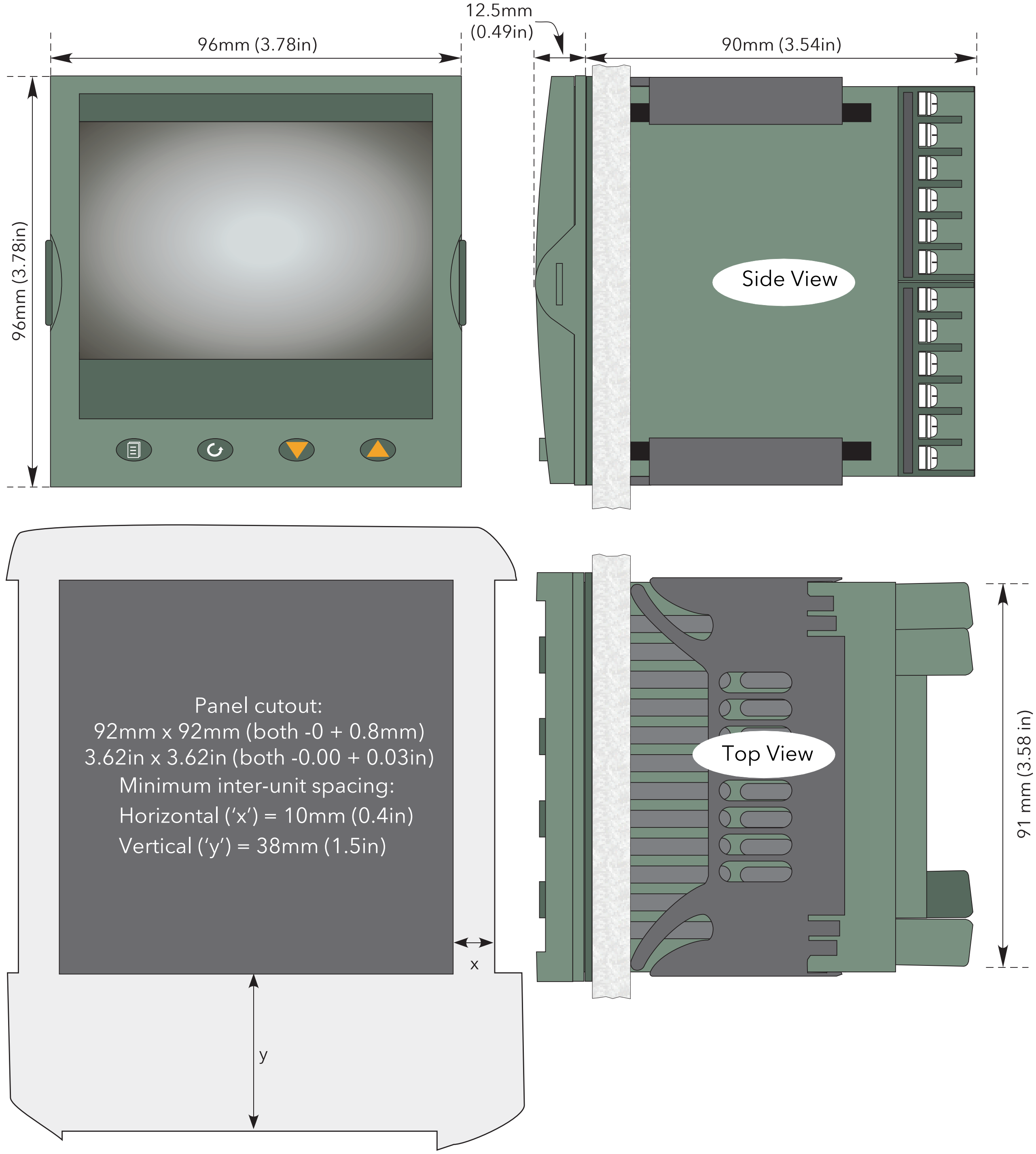



Figure 2.1a Mechanical installation details (standard case)

## 2.1 MECHANICAL INSTALLATION (Cont.)

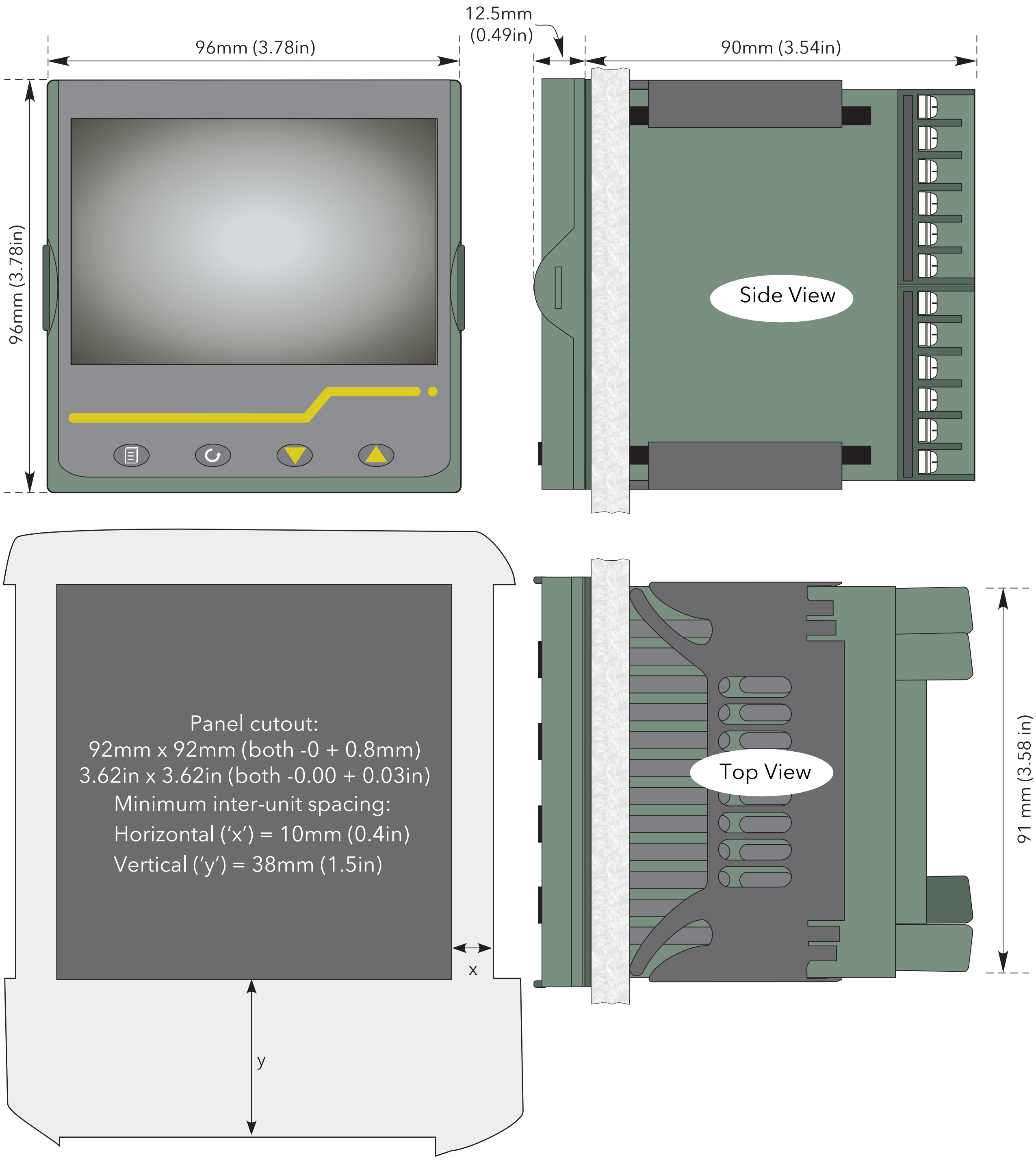



Figure 2.1b Mechanical installation details (wash down case option)

## 2.2 ELECTRICAL INSTALLATION

Figure 2.2 shows the locations of the various user terminations along with signal and supply wiring pinouts.

### 2.2.1 Termination details

The screw terminals accept single wires in the range 0.21 to 2.08 $mm^2$ (24 to 14 AWG) inclusive, or two wires each in the range 0.21 to 1.31 $mm^2$ (24 to 16 AWG) inclusive.

Screw terminals should be tightened to a torque not exceeding 0.4Nm (3.54 lb in)

## 2.2 ELECTRICAL INSTALLATION (Cont.)

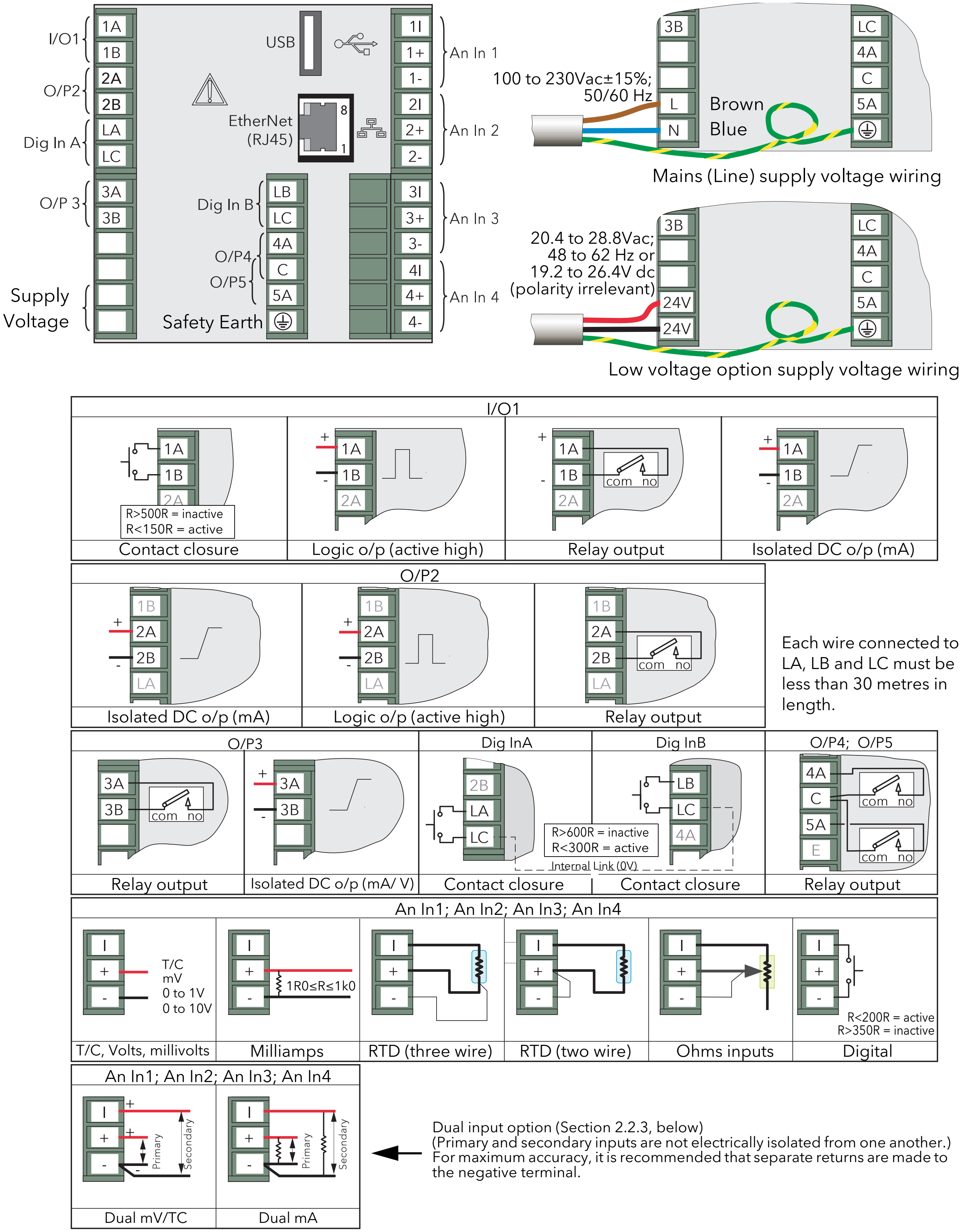



Figure 2.2 Connector locations and pinouts (rear panel)

### 2.2.2 Low Voltage option

This option allows the use of a low voltage ac or dc 24 V supply. The specification in Appendix A gives full details. The polarity of the dc supply connection is not important - it may be connected either way round.

### 2.2.3 Dual input option

This is a cost option, enabled on a channel-by-channel basis by means of entering the relevant password in the 'Feature3 Pass' field in Instrument.Security menu described in section 4.1.6.

For each enabled channel, a pair of thermocouple, mV or mA inputs can be connected to the instrument. These inputs are called 'primary' and 'secondary', and are terminated at the analogue input terminals (An In1 to An In 4) as shown in 'figure 2.2, above. The primary inputs 1 to 4 are assigned to channels 1 to 4, as normal. Each secondary input must be soft wired to a maths channel configured as Operation = 'Copy' if it is to be recorded/displayed/alarmed etc.

> Note: Due to the nature of the input circuit, a large offset may appear for secondary thermocouple inputs. This offset can be removed only by using the input adjust feature described in section 4.1.9. Because of this offset, the dual thermocouple input option is not suitable for AMS2750D applications

Soft wiring is described in Section 7.
Maths channels are described in section 4.5.1.
Channel configuration is described in section 4.4.1.
Input adjust is carried out as described in section 4.1.9

#### SAMPLE RATE

For dual input channels, both primary and secondary sample rate is reduced to 4 Hz (250ms) from the normal 8Hz (125ms).

#### SENSOR BREAK DETECTION

Input sensor break detection is not supported for secondary inputs. The internal circuit acts as a 'pull up' on the secondary input which therefore saturates high in the event of a sensor break.

#### DUAL MILLIAMP INPUT OFFSET CORRECTION

If 'Dual mA' is selected as input type, then an automatic offset correction will be made, according to the shunt value entered in channel configuration.

#### INPUT RANGE LIMITATION

There is no 10V range associated with the secondary input. Any input greater than +2V or less than -2V is deemed to be 'bad range'.

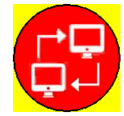
### 2.2.4 Modbus Master communications

The master instrument can be connected directly to up to two slaves using standard ethernet network cable either directly (single slave only) or via a hub or switch (one or two slaves). In either case, 'straight through' or 'crossover' cable may be used. The cable is terminated at the RJ45 socket at the rear of the unit.

### 2.2.5 EtherNet/IP

The Client and Server are connected in the same way as described above for Modbus Master communications, except that there can be only one client and one server.

# 3 OPERATION



On power up a default or custom (section 4.1.5) splash screen appears and remains visible whilst the unit is initialising. If during this process a network broadcast storm is detected, the unit stops, displaying a network failure icon until the broadcast storm has cleared, after which the initialisation process resumes.

## 3.1 INTRODUCTION

The operator interface consists of a display screen and four push buttons.

### 3.1.1 Display screen

The display screen is used both to display channel information (in one of a number of display modes), and to display the various configuration screens which allow the user to setup the recorder to display the required channels, to set up alarms and so on. Display modes are described in section 3.4, below; configuration is described in section 4.

In display mode, the screen is split horizontally into three areas (figure 3.1.1)

1. a faceplate giving channel details.
2. the main display screen showing channel traces etc.
3. the status area, displaying instrument name, the current time and date and any system icons.

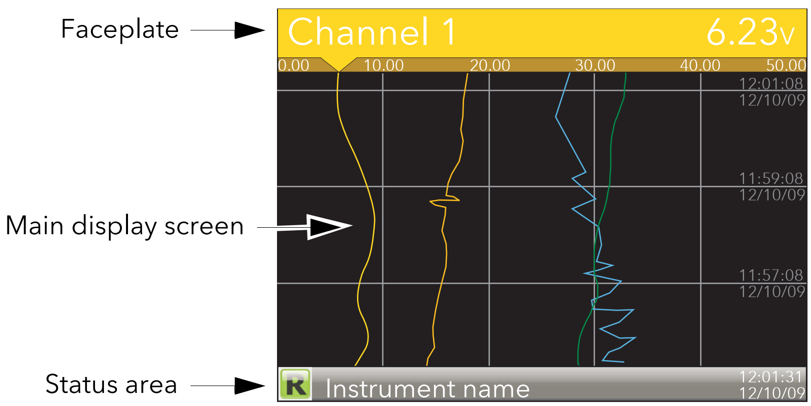



Figure 3.1.1 Display mode screen (vertical trend)

In configuration mode, the entire display screen is devoted to the selected configuration menu.

### 3.1.2 Navigation pushbuttons

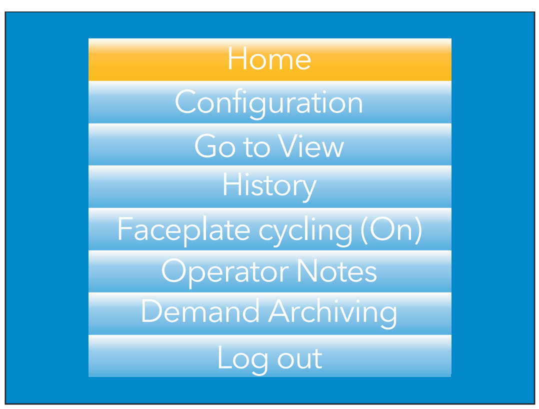



Figure 3.1.2 Top level menu (Engineer level access)

There are four navigation buttons, called 'Page', 'Scroll', 'Lower' and 'Raise' located below the screen. The general properties of these buttons are described in the remainder of this section, but some have additional, context sensitive functions, which, for the sake of clarity are not described here but in the relevant sections (e.g. 'Message summary') of the manual.

### 3.1.2 NAVIGATION PUSHBUTTONS (Cont.)

**PAGE BUTTON**

From any non-configuration page, pressing this push button causes the top level menu (figure 3.1.2) to appear. The figure shows the menu for a user logged in with 'Engineer' level access. Other access levels may have fewer menu items.

Within configuration pages, the Scroll button can be used as an enter key to select lower menu levels. In such cases the page button is used to reverse this action, moving the user up one menu level per operation.

**SCROLL BUTTON**

From trending pages, operation of the scroll push-button scrolls through the channels enabled in the group. The Faceplate cycling 'Off' selection can be used to keep a particular channel permanently displayed, and the scroll pushbuttons can then be used to select channels manually.

In configuration pages, the scroll key operates as an 'enter' key to enter the next menu level associated with the highlighted item. Once the lowest menu level is reached, operation of the scroll key allows the value of the selected item to be edited by the relevant means (for example, the raise/lower keys, or a keyboard entry).

The 'Page' key is used to move the user back up the menu structure, until the top level menu is reached, when the scroll key can be used again to return to the Home page.

The scroll button is also used to initiate user wiring as described in section 7

**RAISE/LOWER BUTTONS**

Within trending displays, the Raise and Lower keys can be used to scroll through the enabled display modes in the sequence: vertical trend, horizontal trend, vertical bargraph, horizontal bargraph, numeric, vertical trend... and so on.

Within configuration pages, these pushbuttons act as cursor keys, allowing, for example, the user to highlight menu items for selection using the scroll button, and in many cases allowing the user to select one from a number of alternative values within menu items. Theses keys are also used to navigate through the virtual keyboards (section 3.6) and number pads used to enter text or numeric strings.

### 3.1.3 On screen help

The top level configuration menu includes contextual help text on the right-hand half of the screen. Mostly this text fits within on screen height. Where this is not the case, the text can be moved up or down the screen by holding the Page button operated whilst using the up and down arrows to move the text.

The down arrow moves the text upwards on the screen; the up arrow moves it downwards.

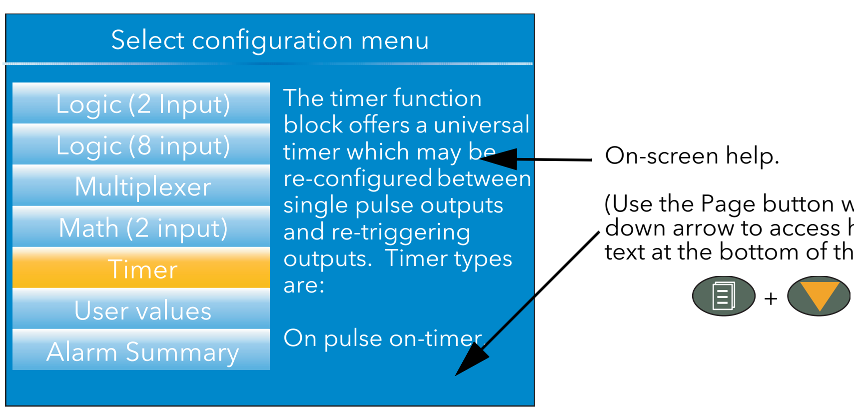



Figure 3.1.3 On-screen help (typical)

## 3.2 PROCESS VARIABLE DISPLAY

As discussed above, the operator interface consists of a display screen and associated push buttons. The display screen shows process variables in one of a number of formats, or operational details (notes or alarm history for example), or configuration details for use in setting up the recorder to produce the required displays and history formats. The remainder of section three discusses the process variable displays, alarm displays and so on; configuration details are to be found in section 4.

Note: Some of the items below can be selected for use only by users with a suitable permission level as set up in the 'Instrument' 'Security' menu described in section 4.1.6

Figure 3.2 below, depicts a typical trend display and gives details of the various areas of the display page.

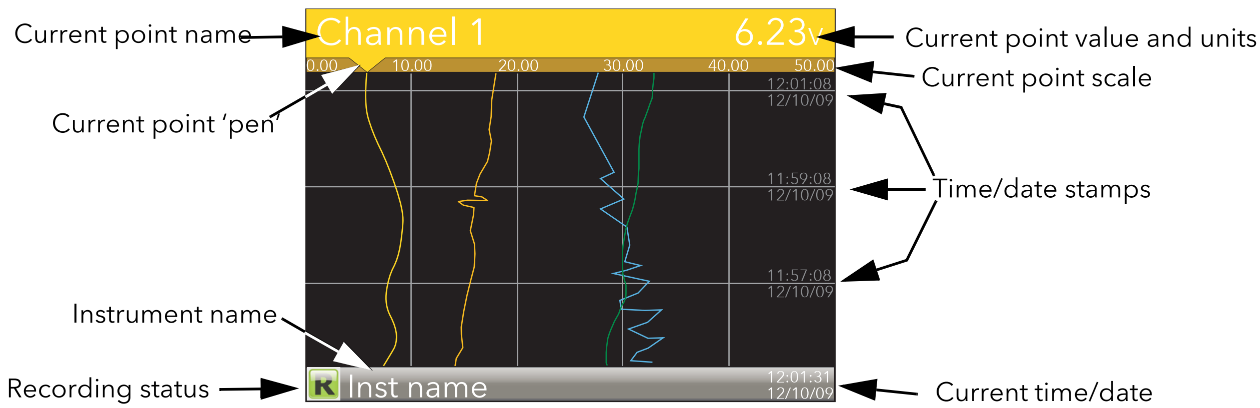



Figure 3.2 Typical display screen (Vertical trend)

Figure 3.2 shows a vertical trend page. Operating the Raise/Lower push-buttons allows the user to scroll through the other display modes: Horizontal trend, Vertical bargraph, horizontal bargraph, numeric, vertical trend... and so on. All these display modes are described in section 3.4, below.

A display mode can also be selected from the Top level menu 'Go To View' item which appears when the 'Page' key is operated.

The scroll button can be used to scroll through the points in the group, overriding the 'Faceplate Cycling' on or off selection

### 3.2.1 Alarm icons

Notes:

1. A full discussion of alarms is given in the Channel Configuration section of this manual (section 4.4.3)
2. Trigger alarms do not display threshold marks or bars, or faceplate symbols

The alarm icons shown below appear in some display modes. The icons on a channel faceplate show the status of that channel's alarm(s), as follows:

| | |
|---|---|
| Icon is flashing | alarm is active but unacknowledged or it is an Auto alarm which is no longer active but which has not been acknowledged |
| Icon steadily illuminated | the alarm is active and has been acknowledged. |

Alarm thresholds and deviation alarm bars appear for horizontal and vertical trend modes. For deviation bars, the bar stretches from (Reference - Deviation) to (Reference + Deviation). Vertical and Horizontal bargraph modes display only absolute alarm symbols.

### 3.2.1 ALARM ICONS (Cont.)

| | |
|---|---|
| | Absolute High |
| | Absolute Low |
| | Deviation High |
| | Deviation Low |
| | Deviation Band |
| | Rising Rate of change |
| | Falling Rate of change |
| | Digital High |
| | Digital Low |

Table 3.2.1 Alarm icons

### 3.2.2 Status bar Icons

The following items can appear in a dedicated window immediately to the left of the time and date, at the bottom right-hand corner of the display. The width of this window expands as the number of icons increases, and the instrument name is truncated, as necessary, to make room.

**SYSTEM ALARMS**

This indicator appears, flashing, if any one or more of the alarms listed below is active. The System Alarms summary page (accessed from 'Go to View in the top level menu) allows the user to view such system alarms as are active. It is not possible to 'acknowledge' system alarms

| | |
|---|---|
| Archive Disabled | An unattended archiving strategy has temporarily been disabled. |
| Archiving Failed | An unattended archiving strategy has failed to complete. |
| Archiving Timeout | A configured archiving strategy has timed out. |
| Battery failure | Indicates that the battery is approaching the end of its useful life, or that it is missing or is completely exhausted. Immediate battery replacement is recommended (Appendix C; section C1). |
| Broadcast Storm detected | Networking is limited until the storm has passed. |
| Clock failure | The internal clock was found to be corrupt at power up, or that the time has never been set. Time is forced to 00:00 1/1/1900. Can be caused by battery failure, in which case a battery failure message appears. The error is cleared by setting the time and date. |
| Channel error | Indicates a hardware failure in the channel circuit or in the internal cold junction temperature measurement. |
| Database failure | Corrupted EEPROM or flash memory. |
| DHCP Server failure | For units with 'IP Type' set to 'DHCP' (Network.Interface configuration) this alarm occurs if the instrument is unable to obtain an IP address from the server. |
| FTP Archiving file lost | A file has been deleted that had not yet been archived. Possible causes: Communications with the server could not be established,; archive is disabled; archive rate too slow. |
| FTP Archiving to slow | The archive rate is too slow to prevent the internal memory from overflowing. The recorder effectively switches to 'Automatic' (Section 4.2.2) to ensure that data is not lost. |

(Continued)

### 3.2.2 STATUS BAR ICONS (Cont.)

| | |
|---|---|
| FTP Primary Server Failure | This error occurs if the recorder fails to establish connection with the primary server, after two attempts. After the second attempt fails, the recorder attempts to establish connection with the secondary server instead. Primary and secondary server details are entered in the Network.Archiving area of configuration (Section 4.2.2). |
| FTP Secondary Server Failure | This error occurs if the recorder fails to establish connection with the secondary server, after two attempts. Primary and secondary server details are entered in the Network.Archiving area of configuration (section 4.2.2). |
| Maths channel failure | Appears if, for example, the divisor of a divide function is zero. |
| Media archiving file lost | A file has been deleted that had not yet been archived. Possible causes: Memory stick missing, full or write protected; archiving has been disabled; archiving rate too slow. |
| Media archiving to slow | The archive rate is too slow to prevent the internal memory from overflowing. The recorder effectively switches to 'Automatic' (Section 4.2.2) to ensure that data is not lost. |
| Media full | Archive storage device is full. The alarm becomes active only when an archive is in progress. |
| Media missing | No archive storage device present when archive attempted. |
| Non-volatile memory failure | RAM copy of non-volatile parameters is corrupted. |
| Non-volatile Write Frequency warning | One or more parameters are being written frequently to non-volatile memory. If this continues, it may lead to 'memory depletion' (i.e. the memory will no longer be able to store values correctly). A common cause of this problem is frequent writes over Modbus comms. |
| Recording failure (message) | Message explains reason for failure. |
| USB overcurrent | USB power fault - too much current (i.e. >100mA) is being drawn by a USB device. |
| Wiring failure | The user wiring has failed to verify, i.e. one or more wires has been detected that does not have both a source and a destination defined. This may be the result, for example, of power loss during a download from iTools. |

**CHANNEL ALARM**

This indicator appears if any channel (including channels not in the display group) is in an alarm state. The symbol is illuminated continuously if all alarms are acknowledged or flashes if any one or more alarms is unacknowledged. Alarms are acknowledged from the Root menu 'Alarm summary' item as described in section 3.3.3 or in the Channel configuration area (Section 4.4.3) if the user's access permission is appropriate.

**USB**

This icon appears whenever a memory stick (max. capacity 8GB) or other supported USB device (section 8) is plugged into the USB port at the rear of the recorder.

When data transfer is in progress between the instrument and the memory stick, the icon changes to a 'busy' version.

**CAUTION**

The Memory stick must not be removed while archiving (demand or automatic) is in progress, as to do so may irreparably damage the file system of the memory stick, rendering it unusable. It is recommended that all archiving be suspended before the memory stick is removed.

Memory stick fitted

Transfer in progress

**FTP ICON**

The FTP icon appears whenever transfer activity is taking place.

### 3.2.2 STATUS BAR ICONS (Cont.)

#### RECORD ICON

One of four icons appears at the bottom left corner of the display to indicate recording status.

Record

This indicates that the recorder is recording the items selected in the Group Recording area of configuration (section 4.3).

Stopped

This means that 'Enable' has been set to 'no' in the Group Recording area of configuration (section 4.3). Trending is not affected.

Paused (Suspended)

This means that recording has been paused by a wire to the Suspend parameter (Group Recording area of configuration (section 4.3)) going true (high). Trending is not affected.

In Configuration

The recorder has been placed in configuration mode either at the user interface, or via iTools. Recording is stopped until the recorder is no longer in configuration mode. For each non-recording state (Stopped, Paused or In Configuration). A new history file is created when the unit comes out of configuration mode.

Note: For recording to be enabled, configuration status must be 'logged out' both at the instrument and at iTools.

#### MESSAGE ICON

This 'envelope' icon appears when a message is generated and it remains on display until the Message Summary is accessed, when it is removed from the display until the next new message is generated.

#### AUTOTUNE ICON

For instruments fitted with the Loop option, this symbol appears during the Autotune process.

### 3.2.3 Breaks in recording

Breaks in recording can be caused by the unit being powered down, by the user entering configuration mode or when the recorder time is changed manually. In vertical and horizontal trend modes, a line is drawn across the width/height of the chart to indicate that recording has been interrupted.

On power up, a red line is drawn across the chart. In 'History', if messages are enabled the message:

`Date Time System power up`

is printed on the chart, together with the configuration and security revisions.

On exiting configuration mode, a blue line is drawn on the chart and in 'History', if messages are enabled, the messages:

`Date Time Logged out.`

`Date Time Config Revision: N was N-1` (assuming a configuration change was made)

`Date Time Logged in as: Engineer`

appear on the chart.

When the instrument time is changed (manually - not through daylight saving action) a green line is drawn on the chart and in 'History', if messages are enabled, the message:

`Date Time Time/Date changed`

appears on the chart.

## 3.3 TOP LEVEL MENU

This menu appears when the page key is operated from any non-configuration page. The menu items displayed depend on the access permission of the user. One of the menu items is highlighted, and if the scroll key is operated, then it is the highlighted item that is 'entered'.

Figure 3.3 shows the top level menu for Engineer level access.

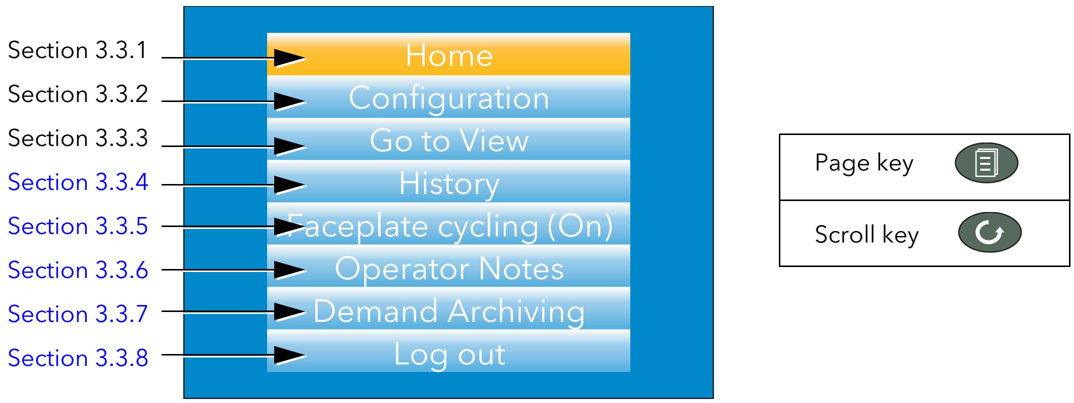



Figure 3.3 Top level menu

### 3.3.1 Home

Operating the scroll key whilst 'Home' is highlighted causes a return to the 'Home' page. By default, this is the vertical trend mode, but the mode can be changed in 'Instrument.Display' configuration (section 4.1.3)

### 3.3.2 Configuration

Operating the down arrow key highlights the 'Configuration' item. Operating the Scroll key enters the configuration submenu described in section 4 of this manual.

Note: 'Configuration' appears only if the user has an appropriate access level.

### 3.3.3 Go to View

Operating the scroll key whilst the 'Go to view' item is highlighted, calls the Go to view submenu (figure 3.3.3a). This allows the user to view channel alarms, system alarms, messages or to select a different display mode.

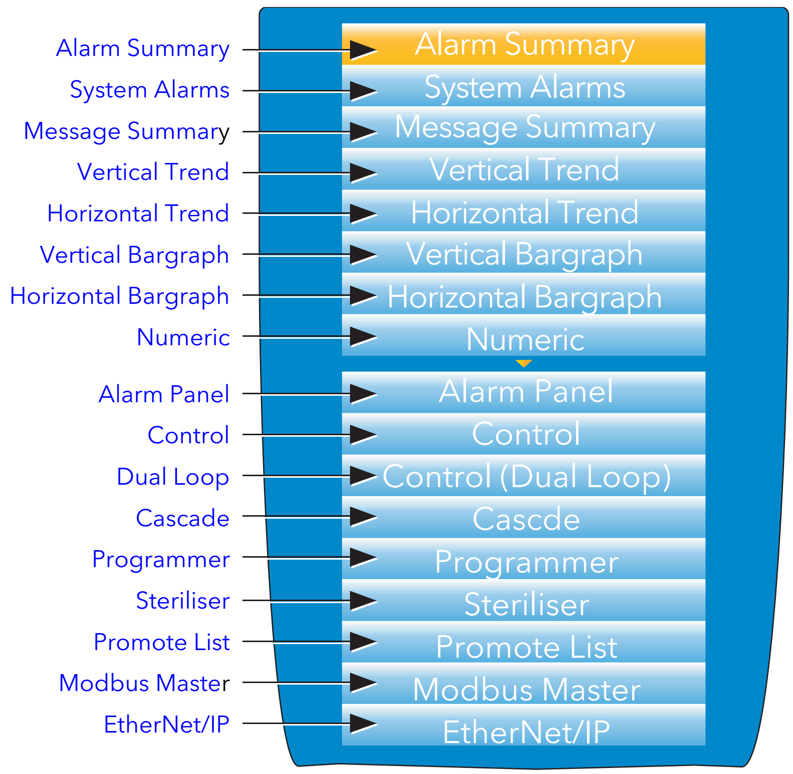



Figure 3.3.3a Go to view submenu

Notes:
1. If an option (e.g. 'Steriliser') is not fitted, its display mode does not appear in the list.
2. Some display modes must be enabled in Instrument.View configuration (section 4.1.3) before they become available.

### 3.3.3 GO TO VIEW (Cont.)

**ALARM SUMMARY**

For each active alarm, this page displays the channel identifier with alarm number (e.g. C1(2) = channel 1; alarm 2), the channel descriptor, the alarm threshold the current process value and an alarm type symbol.

To return to the top level menu, operate the Page key.

Notes:
1. The background colour to the channel ID is the same as that chosen for the channel.
2. A prefix 'C' in the channel ID means that this is a measuring channel; A prefix 'V' means that this is a virtual channel (i.e. a totaliser, counter or maths channel)

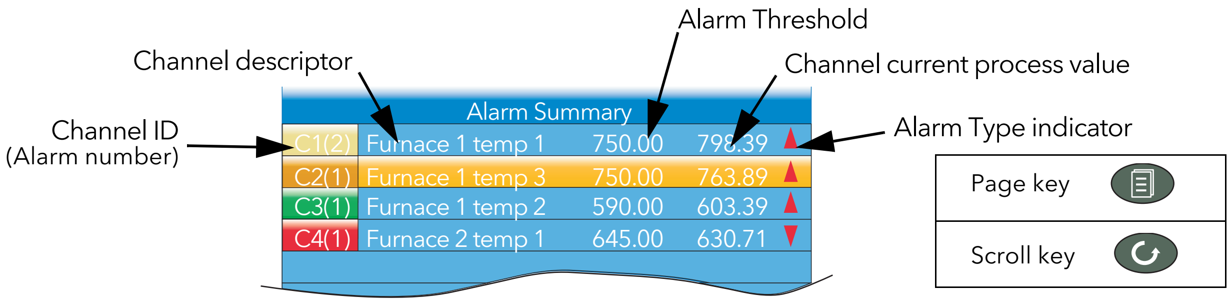
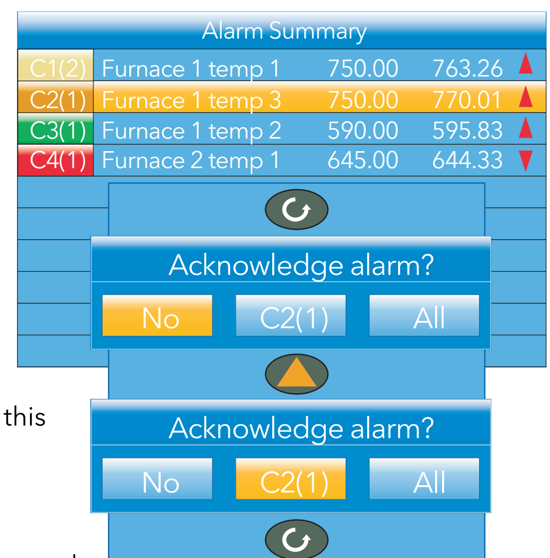



Figure 3.3.3b Alarm summary page with acknowledge confirmation display

ALARM ACKNOWLEDGEMENT

To acknowledge an alarm from this view:

1. Use the up and down arrows to highlight the required alarm




2. Operate the scroll button. The 'Acknowledge alarm' window appears.

3. Use the up arrow to highlight the relevant field (C2(1) in this example), or 'All' if all alarms are to be acknowledged.

4. Operate the scroll key to confirm. If the alarm fails to respond, this may be due to the fact that it has been configured as a 'Manual' alarm, and the trigger has not yet returned to a 'safe' (non-alarm) state, or it could be that the instrument is in a logged out state.

**SYSTEM ALARMS**

Operating the scroll button whilst the 'System Alarms' field is highlighted displays a list of all currently active system alarms. Section 3.2.2 contains a list of system alarms and their interpretations. To return to the top level menu, operate the Page key.

A further operation of the scroll button displays a 'Help Information' page, giving the reason for the highlighted alarm.

Operate the scroll button again to return to the system alarm display.

### 3.3.3 GO TO VIEW (Cont.)

#### MESSAGE SUMMARY

Operating the scroll key whilst the 'Message summary' field is highlighted displays the10 most recent messages.

Operating the scroll key whilst a message is highlighted shows the selected message in more detail (and using the up/down keys allows the other messages to be scrolled through). Whilst in this mode, operating the scroll key again, allows the user to choose to jump to the message's location in trend history mode (section 3.5) or to return to the summary page.

By default, the interface is set up such that:

1. all message types are included
2. the up and down arrow keys cause the highlighted selection to move up or down by one message at a time.

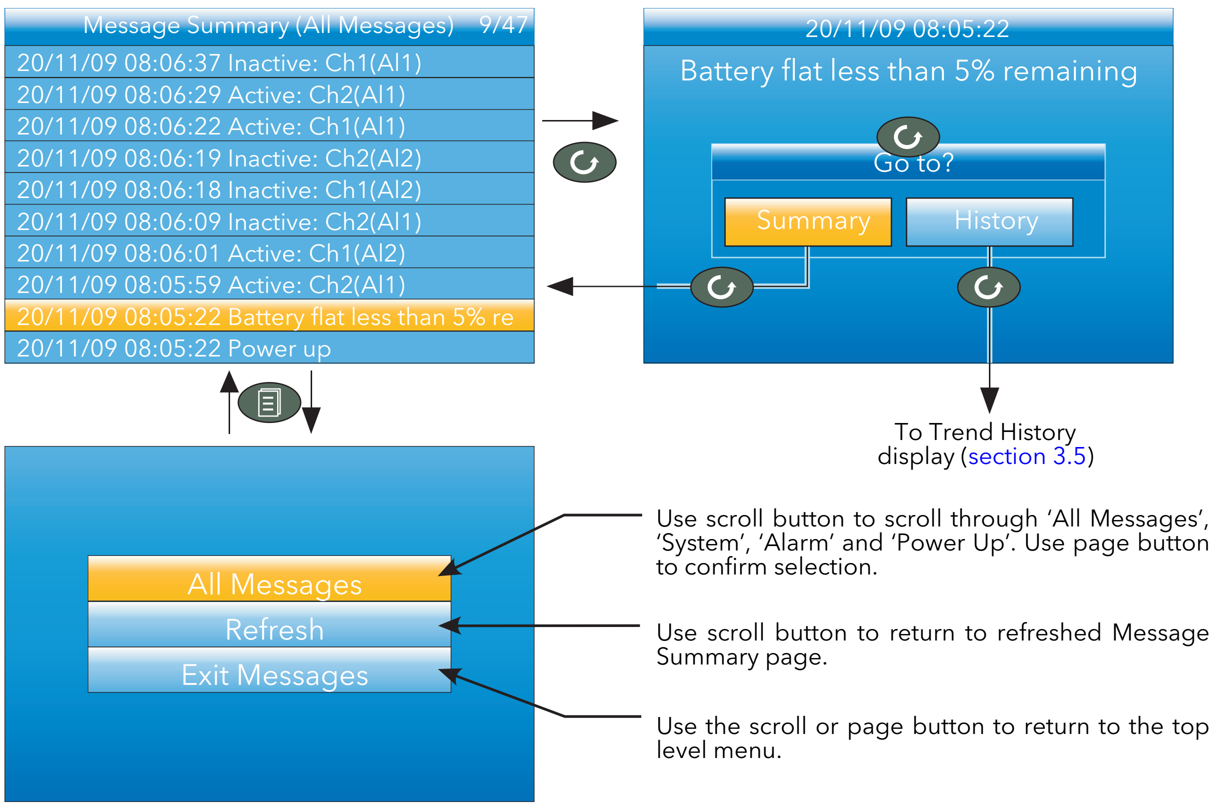
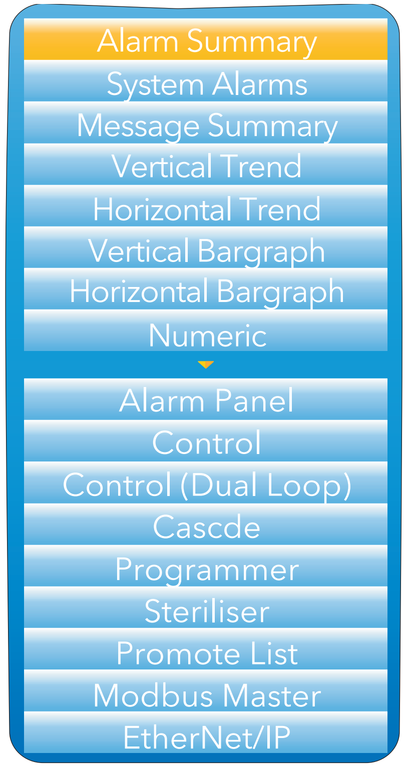



Figure 3.3.3c Message summary features

MESSAGE FILTERS

| | |
|---|---|
| All Messages | Causes all messages to be displayed on the screen. |
| System | Shows only system alarms |
| Alarm | Shows only channel alarms |
| Power up | Shows only power up messages |
| Login/out | Limits the display to Log in and Log out events. |

### 3.3.3 GO TO VIEW (Cont.)




#### DISPLAY MODE SELECTION

Use the up/down arrow buttons to highlight the required display mode. Once the required display mode is highlighted, operation of the scroll button causes the recorder to leave the 'Go to View' menu and to display channel values in the selected mode. See section 3.4 for a description of the various display modes.

Alternatively the up and down arrow buttons can be used from any of the display modes to cycle through the available modes in the order listed in the figure.

Notes:
1. If an option (e.g. 'Steriliser') is not fitted, its display mode is not available for selection.
2. Some display modes must be enabled in Instrument.Display configuration (section 4.1.3) before they become available.

### 3.3.4 History

This top level menu item allows the user to switch from real-time trending to review mode, where channel values, messages, alarm triggers etc. can be viewed back as far as the last significant configuration change. History mode is fully discussed in section 3.5.

### 3.3.5 Faceplate Cycling on/off

For the purposes of this document the channel whose faceplate is currently displayed and whose 'pen' symbol is visible is called the 'Active' channel.

By default, the recorder scrolls through all the channels in the display group, with each channel becoming the active channel in turn. This top level menu 'Faceplate Cycling' item allows the user to inhibit this scrolling action such that the currently active channel remains active permanently, or until a manual scroll is performed using the scroll button (or until Faceplate Cycling is re-enabled).

'Faceplate Cycling' is highlighted by using the up/down arrow buttons. Once highlighted, the status can be changed from 'On' to 'Off' or *vice-versa* using the scroll button. Operation of the 'Page' button returns the user to the trend display.

### 3.3.6 Operator Notes

This area allows up to 10 notes to be created when logged in as Engineer, using either the text entry techniques described in section 3.6, or 'iTools' described in section 6. Once logged out, operating the scroll button whilst a note is highlighted calls a selection box allowing the user either to send that note to the chart, or to write a Custom Note.

#### CUSTOM NOTE

The Custom Note is written using the text entry techniques described in section 3.6. Once the note is complete, operation of the page button calls a confirmation display. The down arrow is used to highlight 'Yes', and when the scroll key is then operated, the message is sent to the chart. This custom message is not retained for further use, so if it is required on a regular basis, it is suggested that one of the Operator Notes 1 to 10 be configured (Engineer access level required) so that it may be used instead.

Note: Each note can contain up to 100 characters.

### 3.3.7 Demand Archiving

This allows a user, with a high enough access level, to archive a selected portion of the recorder history, either to a 'memory stick' plugged into the USB port at the rear of the recorder (Local Archiving), or to a pc, by means of the FTP protocol (Remote Archiving). The archived data remains in the flash memory of the instrument. When the flash memory is full, new data causes the oldest file(s) to be discarded.

The up and down arrow keys are used to navigate to the required field.

### 3.3.7 DEMAND ARCHIVING (Cont.)

#### ARCHIVE MENU

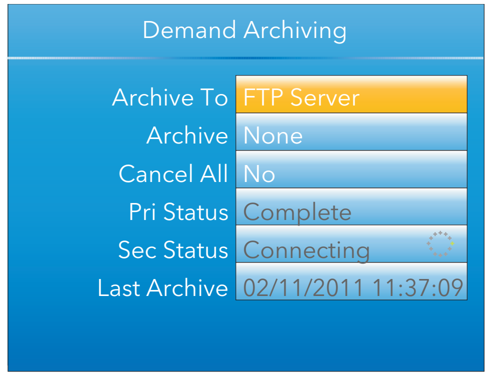




Figure 3.3.7 Demand Archiving menu (Local Archiving on left; Remote Archiving on right)

Archive To
: With this item highlighted, the scroll button and the up/down arrows can be used to select 'USB' or 'FTP Server'.
For 'USB', the archive will be made to the rear USB memory stick. For 'FTP Server' the archive will be made to the Primary or Secondary server (configured in the Network.Archive area of configuration described in section 4.2.2). For more details about remote archiving, see 'Remote archiving', below.

Archive
: In a similar way, select the archive period:
None: No archiving to take place. (Not editable when logged out)
Last Hour: Archives all files created within the last 60 minutes.
Last Day: Archive all files created in the last 24 hours.
Last Week: Archives all files created in the past seven days.
Last Month: Archives all files created in the past 31 days.
Archive All: Archives all the files in the recorder's history.
Bring To Date: Archives all files created or updated since the 'Last Archive' date and time.

Suspend Schedule
: When set to 'Yes', automatic (scheduled) archiving is stopped, once the transfer of the current file is complete. Suspend Schedule must be set to 'No' again, to restart the suspended archive. Suspend can be used to allow the memory stick to be removed and re-fitted safely.

Cancel All
: When set to 'Yes', this cancels USB archiving activity immediately, or cancels FTP archiving once transfer of the current file (if any) is complete.

Last Archive
: Shows the date and time at which the last archive (demand or automatic) was attempted. If a demand archive is requested, or is in operation when an automatic archive is triggered, the automatic archive takes precedence.

Status
: For Archive to USB only
'Complete' means that no archiving is currently taking place.
'Transferring' indicates that an archiving is in progress. Accompanied by an animated circular display.
'Suspended' means that archiving has been suspended as requested.

PriStatus
: For Archive to FTP Server only, this shows the transfer status between the instrument and the primary host computer.

SecStatus
: For Archive to FTP Server only, this shows the transfer status between the instrument and the secondary host computer.

### 3.3.7 DEMAND ARCHIVING (Cont.)

#### FTP SERVER ARCHIVING

This allows the archiving of recorder files to a remote computer via the RJ45 type connector at the rear of the recorder, either directly or via a network.

In order to carry out a successful transfer:

1. Details of the remote host must be entered in the Network.Archive area of configuration (section 4.2.2).
2. The remote computer must be set up as an FTP server. Help from the user's IT department may be necessary in order to achieve this. Appendix C, Section C2 to this manual suggests one way, using Filezilla.
3. The remote computer must also be set up to respond to 'pings'. This is because the instrument pings the host whilst establishing connection, and if it does not receive a response the archive attempt fails.

When accessing files using Microsoft® Internet Explorer, the address (URL) field can be in one of two formats:

1. ftp://<instrument IP address>. This allows a user to log in as the anonymous user (if the recorder has any account with the user name set to 'anonymous' with a blank password.
2. ftp://<user name>:<password>@<instrument IP address> to log in as a specific user.

For IE5 users, Microsoft® Internet Explorer displays, by default, history files only. To quit the history folder, either uncheck the Tools/Internet Options/Advanced/Browsing/'Enable folder view for FTP sites' option, or check the Tools/Internet Options/Advanced/Browsing/'Use Web based FTP' option.

#### REVIEW SOFTWARE

'Review' is a proprietary software package which allows the user to extract 'archive' data from one or more suitable instruments* and to present this data on a host computer, as if on a chart, or as a spreadsheet. The host computer must be set up as an ftp server (see Appendix C section C2 for a description of one way of doing this).

As described in the Review help system, 'Review' allows the user to set up a regular transfer of data (using ftp) from connected instruments into a database on the pc, and then from this database to the chart or spreadsheet. The chart/spreadsheet can be configured to include one or more 'points' from one or all connected instruments (where a 'point' is an umbrella term for channel, totaliser, counter etc.).

It is also possible to archive instrument history files to a memory stick, Compact Flash card etc. (depending on instrument type) and to use this to transfer the data to the pc.

Each type of instrument has its own remote user name and password configuration - for this instrument, the user name and password are both 'history and they are not editable.

*Suitable instruments are connected instruments, the archive files of which have the suffix '.uhh'.

### 3.3.8 Login

Login allows the user to enter a password in order to gain access to areas of the unit's configuration which are not available when the user is logged out.

#### LOGGED OUT ACCESS LEVEL

Logged out mode allows the user to select viewing mode, to view history, to view alarms, to toggle faceplate cycling on and off, to send notes, to suspend/resume USB archiving and to access the login process.

#### OPERATOR ACCESS LEVEL

In addition to the logged out features, Operator access level allows the user to acknowledge alarms, to edit notes and to perform demand archive operations.

By default, no password is required in order to enter Operator level, but a password can be set either at Supervisor level or at Engineer level.

### 3.3.8 LOGIN (Cont.)

#### SUPERVISOR ACCESS LEVEL

In addition to the logged out level function, this access level allows the user to view the recorder's configuration, and to edit some values (such as alarm thresholds). By default, there is no password required to enter Supervisor level, but a password can be set in the Instrument area of configuration, either at Supervisor level or at Engineer level.

#### ENGINEER ACCESS LEVEL

This allows full access to all areas of the recorder configuration. The default password is 100, but this can be edited in the Instrument area of configuration (section 4.1.5).

Note: recording is stopped for as long as the user is logged in at Engineer level, even if the recorder is not being configured. This is indicated by the Record icon at the bottom left corner of the process value display screen being replaced by the Configuration (wrench) icon.

Record icon

Configuration icon

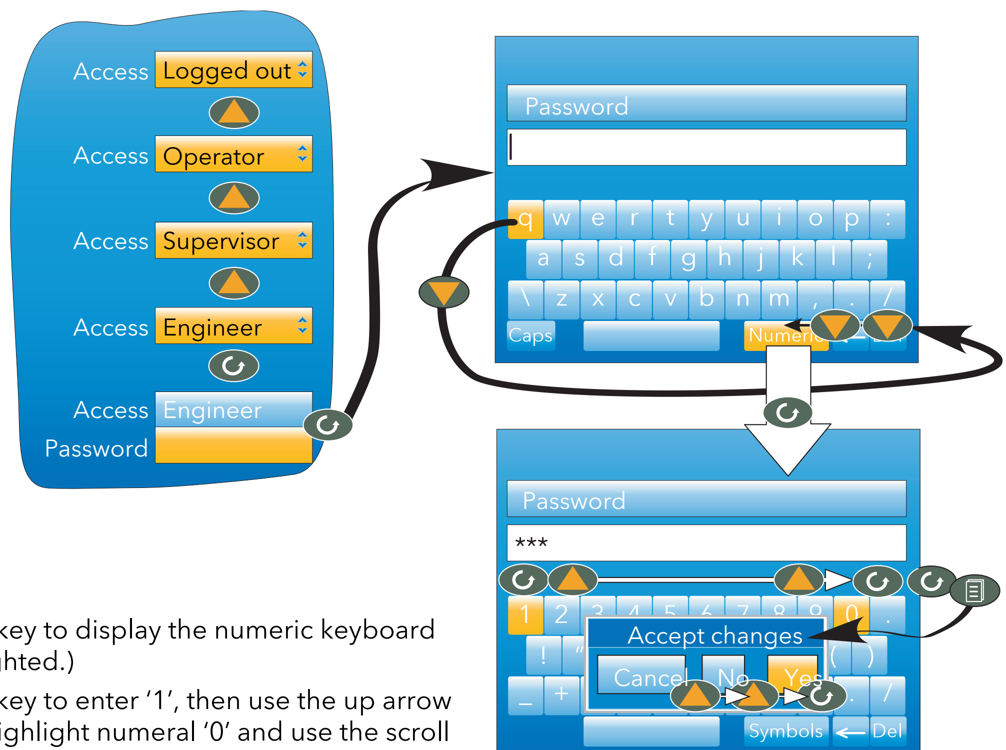
#### LOGIN PROCEDURE

From the top level menu, use the up or down arrow keys as often as necessary in order to highlight 'Login', and then operate the Scroll key to produce the 'Access Logged out' display.

Note: this procedure describes how to login to an access level with a password associated with it. For non-password protected logins, the user needs only to select the required access level, and press the scroll key.

To log in as Engineer (password = 100):

1. Operate the up arrow key three times, to display 'Engineer'.
2. Press the scroll key to call the 'alpha' keyboard, with the letter 'q' highlighted.
3. Use the down arrow key three times to highlight 'Numeric'.
4. Operate the scroll key to display the numeric keyboard (numeral '1' highlighted.)
5. Operate the scroll key to enter '1', then use the up arrow key nine times to highlight numeral '0' and use the scroll key twice to enter '0' 0', completing the password of 100.
6. Use the Page key to call the confirmation display.




7. If the password entry is as required, use the up arrow twice (or the down arrow once) to highlight the word 'Yes' and operate the scroll key to confirm. The top level configuration menu appears. Otherwise, 'Cancel' can be used to clear the entry in order to start again, or 'No' can be used to quit login.

## 3.4 DISPLAY MODES

The following subsections describe the various display modes available to the user. By default, the 'Home' display mode is 'Vertical Trend', but this can be edited as a part of 'Instrument.Display' configuration. This configuration area also allows the user to disable one or more display modes should they not be required.

The current display mode can be chosen either by using the top level menu 'Go to View' item or, from any display mode, by scrolling through the enabled modes using the up or down arrow button.

Details of the various display modes are to be found in the following subsections:

| | |
|---|---|
| Vertical trend ................ section 3.4.1 | Cascade............................................section 3.4.8 |
| Horizontal trend ........... section 3.4.2 | Programmer (inc. future trend)........section 3.4.9 |
| Vertical bargraph .......... section 3.4.3 | Steriliser............................................section 3.4.10 |
| Horizontal bargraph ..... section 3.4.4 | Promote list......................................section 3.4.11 |
| Numeric......................... section 3.4.5 | Modbus Master ................................section 3.4.12 |
| Alarm panel .................. section 3.4.6 | EtherNet/IP.......................................Section 3.4.13 |
| Control loop 1/2........... section 3.4.7 | |

### 3.4.1 Vertical trend

In this mode, channel values are traced as though on a chart rolling downwards (i.e with the latest data at the top). The chart speed, and the number of major divisions are configured in the 'Group.Trend' area of configuration (section 4.3.1). By default, the chart background is black, but this can be changed to white or grey in the 'Instrument' 'Display' area of configuration (section 4.1.3).

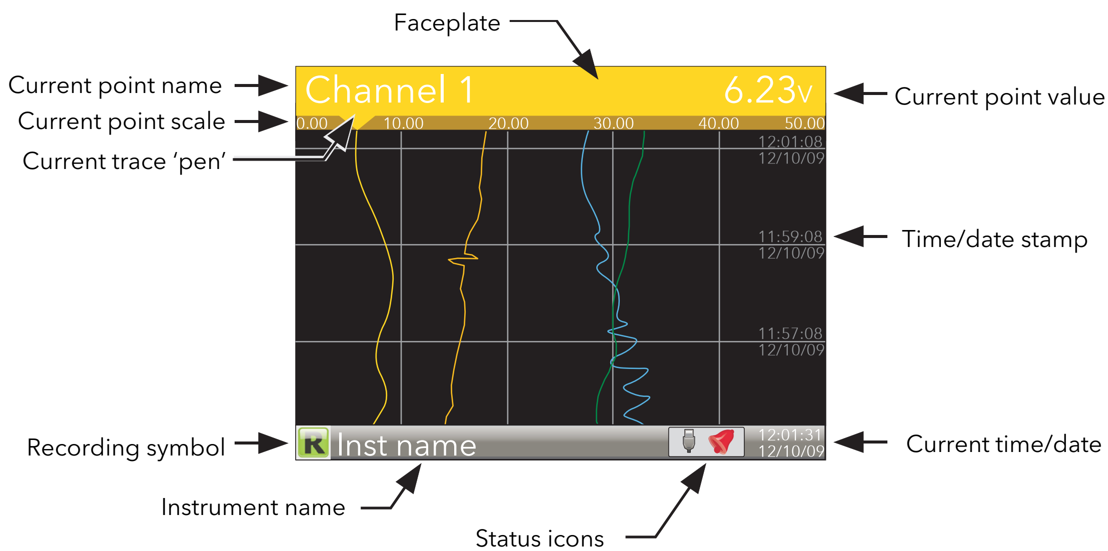



Figure 3.4 Vertical trend mode display elements

One of the channels is said to be the 'current' or 'scale' channel. This channel is identified by its pen icon being displayed, and by the channel descriptor, dynamic value and its scale being displayed on a 'faceplate' across the width of the display, above the chart.

Each channel in the Group becomes the 'current' channel in turn, for approximately five seconds -i.e. the channels are cycled through, starting with the lowest numbered channel. Once the final channel in the Group has ben displayed for five seconds, the first channel is returned-to and the process repeats. This scrolling behaviour can be enabled/disabled from the top level menu 'Faceplate Cycling (Off)' item described in section 3.3.5.

The scroll button can be used to cycle through the channels manually in both Faceplate cycle on and off modes.

Use of the up arrow button causes the next enabled display mode to be entered (default = horizontal trend).

The page key calls the top level menu.

### 3.4.2 Horizontal Trend mode

This view is similar to the vertical trend mode described in section 3.4.1 above, except that the traces are produced horizontally rather than vertically. Initially, as each channel appears, its scale appears at the left edge of the display (as shown below), but in order to show the maximum amount of trend data, the scale is overwritten after a few seconds.

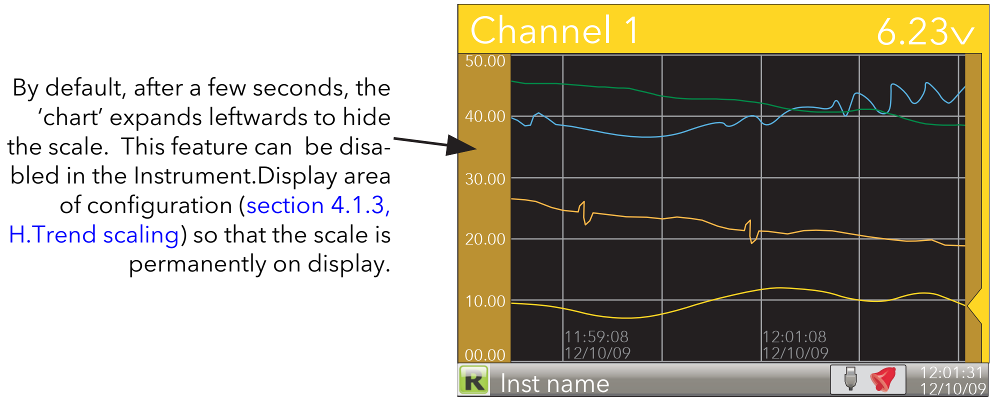



Figure 3.4.2 Horizontal trend display mode

Note: Timestamps appear to the right of the gridline to which they relate

Use of the up arrow button causes the next enabled display mode to be entered (default = vertical bar-graph). Use of the page key calls the top level menu.

### 3.4.3 Vertical Bargraph mode

This display mode shows the channel values as a histogram. Absolute alarm threshold values appear as lines across the bars, grey if the alarm is not triggered; red if the alarm is triggered. Alarm symbols appear for active alarms.

Bargraph widths for four to six channels divide the width of the display screen equally between them. For one and two channels, the width is fixed, and the bars are centred on the screen. Figure 3.4.3 shows some examples (not to the same scale).

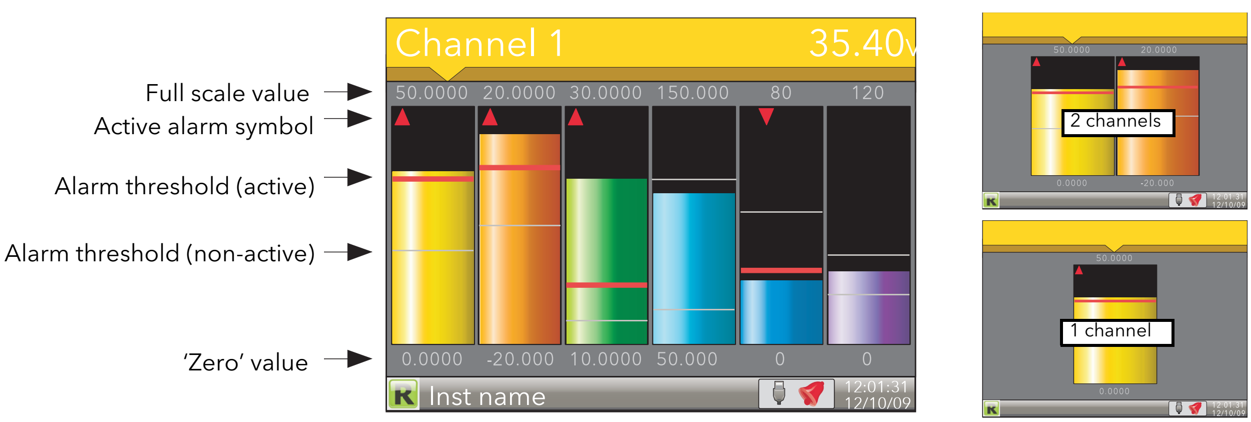



Figure 3.4.3 Vertical bargraph display mode

Use of the up arrow button causes the next enabled display mode to be entered (default = horizontal bar-graph). Use of the page key calls the top level menu.

### 3.4.4 Horizontal Bargraph mode

Similar to the Vertical bargraph mode described in section 3.4.3, above, but includes channel descriptors.

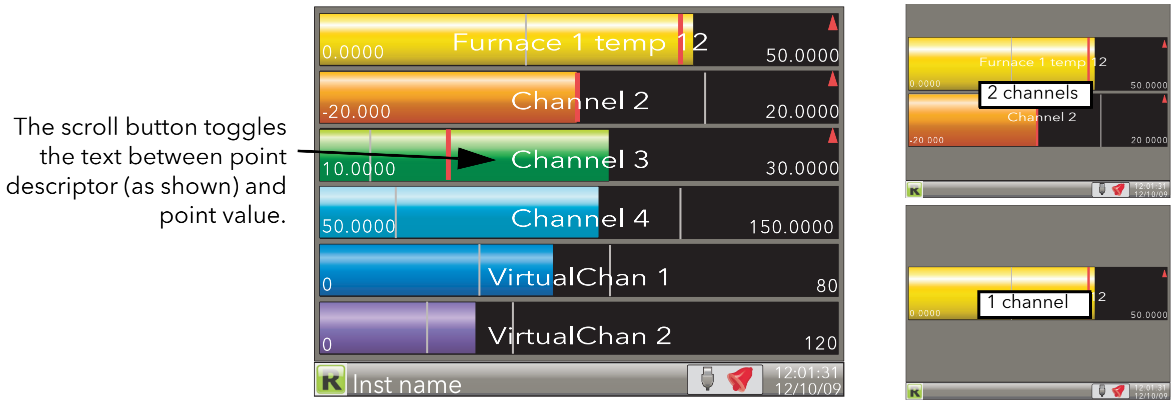



Figure 3.4.4 Horizontal bargraph mode

Use of the up arrow button causes the next enabled display mode to be entered (default = numeric).
Use of the page key calls the top level menu.

### 3.4.5 Numeric mode

Shows the enabled channels' values along with their descriptors and with indications of the type(s) of alarm configured for each channel.

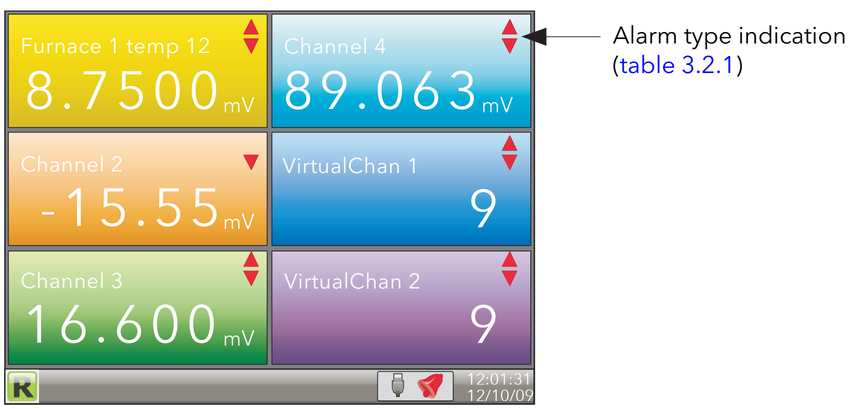



Figure 3.4.5a Numeric display mode (six enabled channels)

The figure above shows an example where the Trend group contains six channels. Figure 3.4.5b shows how the display appears for trend groups with fewer than six channels configured.

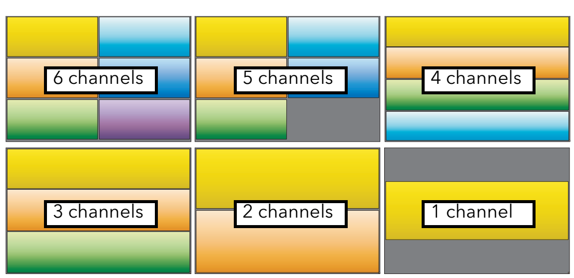



Figure 3.4.5b Display layout for different numbers of channels

The up arrow button returns to the vertical trend display mode; the page key calls the top level menu.

### 3.4.6 Alarm panel

This display appears only if enabled in the Instrument Display configuration (section 4.1.3) Alarm panel mode shows current value and alarm status for each channel enabled in the Trend Group. The status is shown in two ways, by the colour of the relevant bar, and by the alarm status indicators.

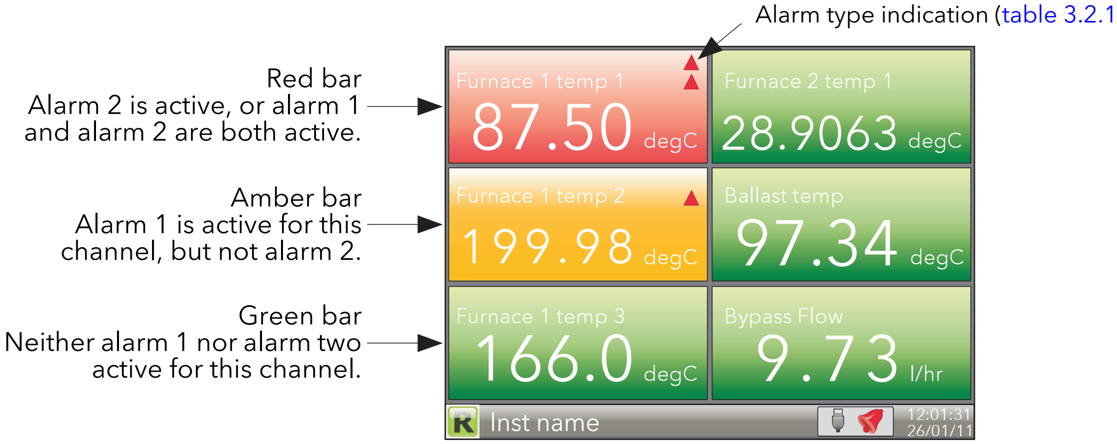



Figure 3.4.6a Alarm panel display (six channels)

The figure above shows an example where the Trend group contains six channels. Figure 3.4.6b shows how the display appears for trend groups with fewer than six channels configured.

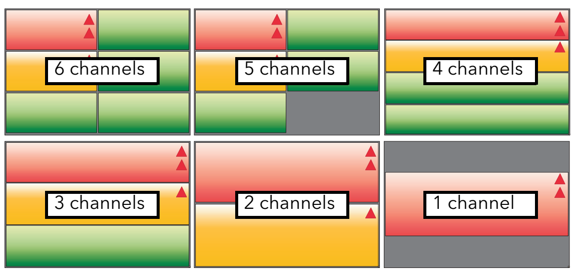



Figure 3.4.6b Alarm panel display layouts for trend groups with fewer than six channels

### 3.4.7 Control Loop1/Loop2

These displays appear only if the controller option is enabled (section 4.1.6).

The loop display modes are interactive, in that the setpoint, the Auto/Manual mode and the Manual Output value can be edited from the user interface. Full configuration is carried out in the Loop setup menus (section 4.6) and a fuller description of control loops is to be found as Appendix B to this manual.

Figure 3.4.7 depicts a single loop display and the dual loop display. The up and down arrow keys are used as normal to scroll through Loop1, Loop2 and Dual loop pages.

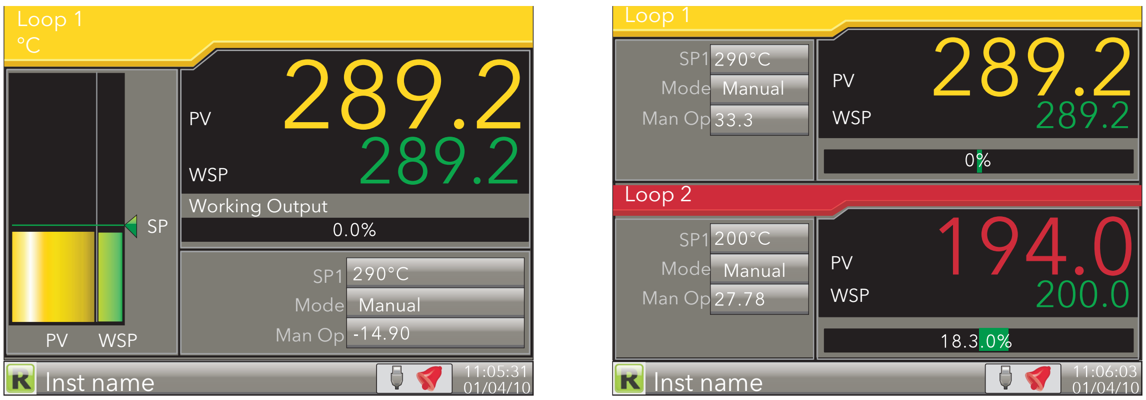



Figure 3.4.7 Loop displays

Note: The colours associated with the loops are those of the channels to which they are wired.

**EDITING TECHNIQUES**

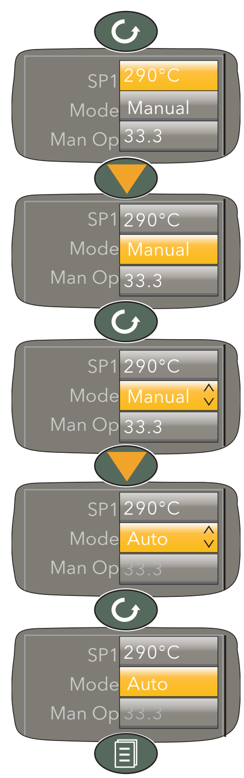
1. With the loop page on display, operate the Scroll key. This highlights the first editable item (SP1). The scroll order includes both loop1 and loop 2 parameters in the dual loop display.

2. Use the up and down arrow keys to select the required field for editing. When the required field is highlighted, operate the scroll key again, to enter edit mode.

3. Use the up/down arrows to edit the current setting.

4. Operate the scroll key to confirm the edit.

5. Select a further parameter for editing, or operate the page key to return to normal operation.




Note: Edit permissions for Setpoint and Auto/Manual are set in the Loop Setup configuration menu (section 4.6.2).

### 3.4.8 Cascade display mode

This display mode appears only if 'Cascade' has been enabled in the Instrument.Display area of configuration (section 4.1.3). See also Advanced Loop configuration (section 4.7).

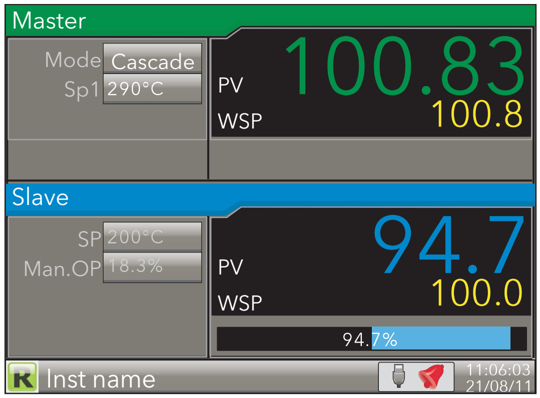



Figure 3.4.8a Cascade display mode

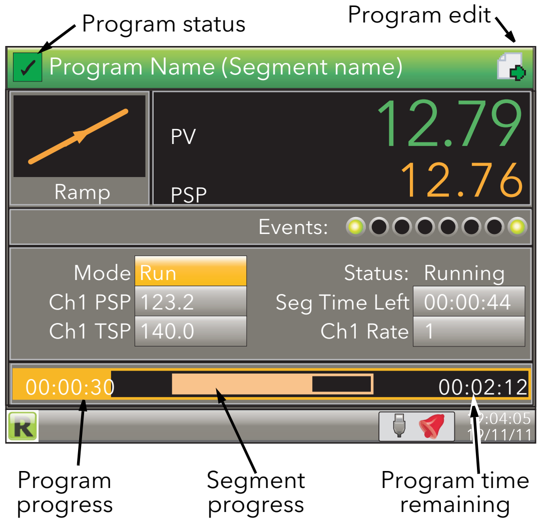
Operating the scroll button highlights the Master 'Mode' field. Operating the scroll button again, enters edit mode allowing the user to use the up/down arrow buttons to scroll through the available modes. Once the required mode appears, a further operation of the scroll button confirms the entry and quits edit mode.

Once out of edit mode, the down arrow key can be used to select Master 'SP1', Slave 'SP' and Slave 'Man OP'. The Mode selected determines how many of these items are editable by the operator.

| | |
|---|---|
| Mode | Cascade: The master loop is in auto mode and provides the slave setpoint. Changing modes causes the slave to switch to the local slave setpoint.<br>Slave: A simple single loop controlling with a local setpoint.<br>Manual: Provides a single manual % power output. |
| SP1 | Setpoint 1 is the primary setpoint of the controller. If the controller is in automatic control mode, then the difference between the setpoint and the process variable (PV) is continuously monitored by the control algorithm. The difference between the two is used to produce an output calculated to bring the PV to the setpoint as quickly as possible without causing overshoot. |
| SP | The slave setpoint, either local (Manual or Slave mode) in which case it can be edited, or supplied by the master loop (Cascade mode), in which case it is not editable. |
| Man.OP | The percentage output power to be applied when in Manual mode (100% = full on; 0% = off). |

Note: The default loop names ('Master' and 'Slave') can be replaced by user-entered strings of up to 10 characters in Advanced Loop Setup configuration (section 4.7.2).

### 3.4.9 Programmer display mode




Single channel


Double channel

Figure 3.4.9a Programmer displays (typical)

This display mode (if enabled - see section 4.1.3) allows the user to monitor the progress of a single or dual-channel setpoint program, and if logged-in as 'Operator*', to reset or run the program. The program itself is created in the Program edit page (described below) and in Programmer configuration (section 4.8 or 'iTools').

*Note: Operator is the default access level - to edit, see ' Prog Mode Access' in section 4.8.3)

The displays contain the following features:

Program name — This is the name of the loaded program. If the program has been modified since being saved, an asterisk (*) appears after the name. Default background colour shown. This colour changes to that assigned to the input channel when this is configured.

Segment name — This is the name of the current segment. If not named in Segment configuration, then the segment number appears instead.

Program status — At the top right hand corner of the display, this can be any one of the following:

The program is running (or ran last time) without any PV 'Alarm' events or user intervention.

The user has intervened in the running of the program, by placing it in 'hold' or 'reset', or by advancing a segment, or by adjusting a duration, target setpoint, ramp rate or time-to-target value.

A PV 'Alarm' Event has activated. A PV 'Alarm' Event is an absolute high/low or a deviation alarm on the PV input.

There is no program loaded, or if a program is loaded, it has not yet run.

Program edit — This icon appears for users with appropriate access permissions, to indicate that setpoint programs can be configured (as described in Program edit, below).

Segment type — For single channel displays, this indicates the type of segment currently being run:

Dwell. The segment value remains constant for the duration of the dwell period.

End (dwell). Displayed on completion of the program. The segment value remains at the final value until reset

End (reset). Displayed on completion of the program. The program resets.

Ramp. The segment value ramps at a fixed rate or over a fixed period to the Target setpoint. Ramp up icon shown; ramp down is similar but inverted.

Step. The segment value switches immediately to the new Target setpoint. Step down shown; step up similar but inverted.

Wait. The segment value remains constant until the wait criteria are satisfied.

### 3.4.9 PROGRAMMER DISPLAY MODE (cont.)

| | |
|---|---|
| PV | The current process value of the signal wired to Ch1(2) PV Input. |
| Ch1(2)PSP | This is the output setpoint from the programmer for the channel. In reset this value tracks the configured servo parameter. |
| Ch1(2)TSP | The channel target setpoint. The target set-point may be edited while the program is in hold (in such cases, for ramp rate segments the time remaining is recalculated. |
| Events | Up to eight events can be configured in the Program Edit page. Any one or more of these events may be deemed to be active for the duration of each individual segment. |
| Mode | Shows the current run mode of the program. If the user has the correct access level, the mode can be set to 'hold', reset' etc. by using the scroll key twice (first to highlight the run mode, then again to enter edit mode) and then using the up/down arrow keys to select the required mode. Run, reset, hold etc. can also be selected by inputs from other parameters, switch inputs etc. |
| Status | Shows the status of the current segment. |
| Ch1 Rate | The channel 1 rate-of-change of segment value for 'Rate' ramp segments. |
| Ch1 Time | Shows the channel 1 duration configured for the segment to ramp, dwell etc. for 'Time' ramp segments. For two-channel programs, see the note below. |
| Seg Time Left | Shows the time that the segment has to run before completion. |
| Program progress | The numerals show program elapsed time, and the bar gives an indication of progress so far. For two-channel programs, see the note below. |
| Segment progress | For each segment as it runs, this gives a visual indication of the proportion of total segment time which has elapsed so far. For two-channel programs, see the note below. |
| Program time remaining | Shows the time remaining until the program completes. For two-channel programs, see the note below. |

Note: For two-channel programs, in 'Hold' mode, the 'program progress', 'segment progress' and 'program time remaining' areas of the display are replaced by 'Ch1 Time' and 'Ch2 Time', as shown below.

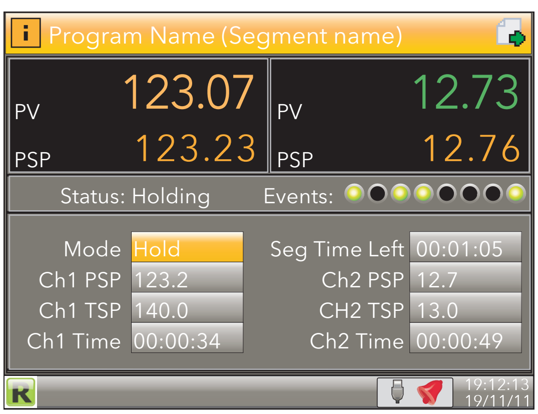



Figure 3.4.9b two channel program in Hold mode

### 3.4.9 PROGRAMMER DISPLAY MODE (cont.)

#### PROGRAM RUN/RESET/HOLD

Programs can be controlled by users with the correct access level (defined in Programmer configuration - section 4.24). The display page is placed in edit mode by operation of the scroll key ('Mode' highlights). A second operation of the scroll key followed by operation of the up/down arrows allows the user to select 'Run', 'Hold' or 'Reset'. A further operation of the scroll key initiates the selected action.

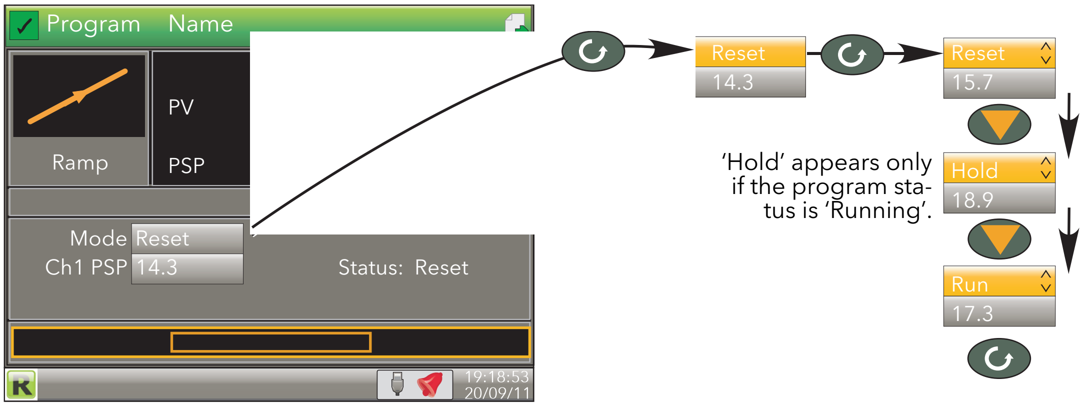


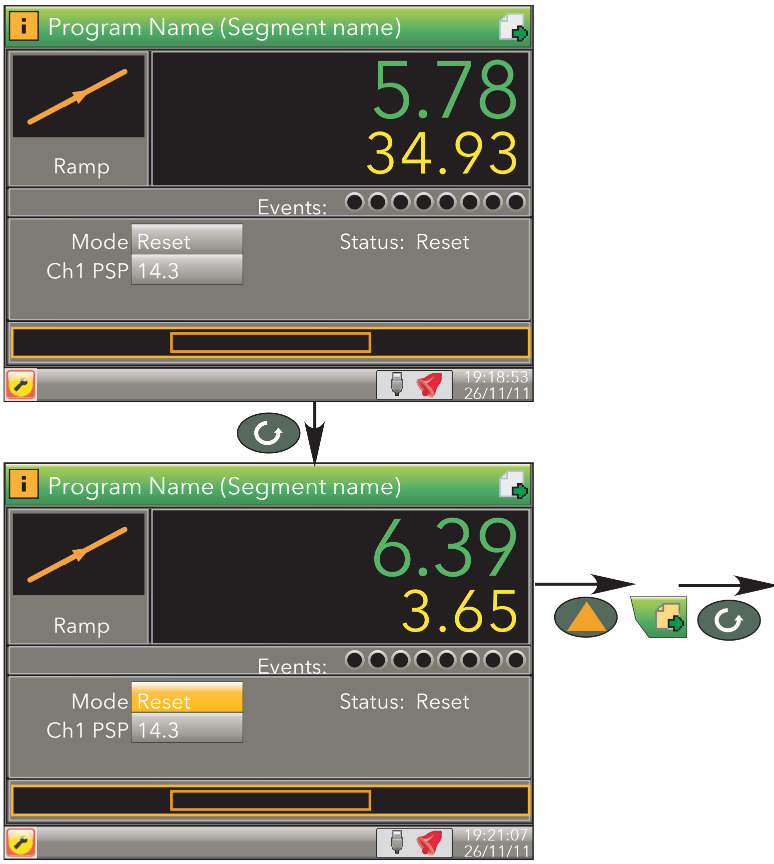

Figure 3.4.9c Setting the Mode

Notes:
1. These functions can also be carried out by wiring relevant inputs to the 'Run', 'Hold' or 'Reset' parameters in Programmer configuration (section 4.8).
2. The user must have either 'Logged off', 'Operator' or 'Supervisor' level access as defined in 'Prog Mode Access' in the Programmer.Setup menu described in section 4.8.3 . The program cannot run if the unit is logged into at 'Engineer' level.

### 3.4.9 PROGRAMMER DISPLAY MODE (cont.)

#### PROGRAM EDITING

The program edit page is accessed by operating the scroll button once to highlight the Mode, then using the up arrow key to highlight the page symbol at the top right hand corner of the display and then the scroll button again to enter the program editor.




By default, Program Edit is available only to users with Supervisor or Engineer level access. The required access level can be edited in Programmer.Set Up configuration as described in section 4.8.3.

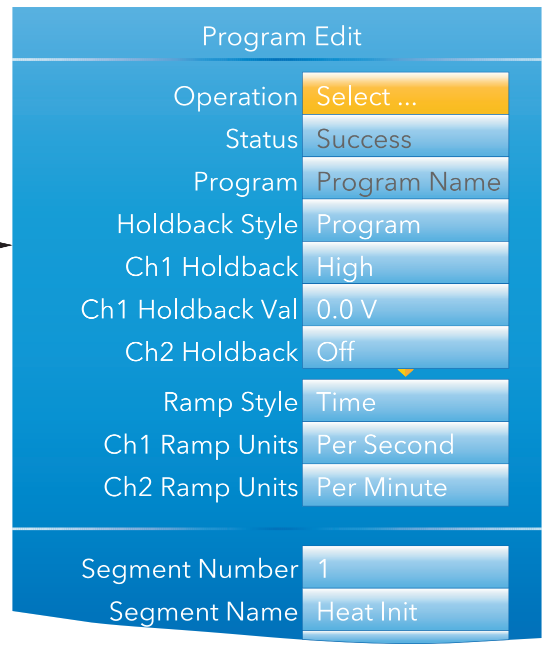



Figure 3.4.9d Access to the program editor

As can be seen from the figure above, the initial Program Edit page is divided into two areas - the top part contains program details; the lower part (figure 3.4.9f) contains individual segment details. The editable items that appear in the program details area depend on the features enabled in the Programmer Features configuration menu (section 4.8.1).

Note: Access to some program operations is restricted to users with the correct access level, as defined in the 'Prog Mode Access', the 'Prog Edit Access' and the 'Prog Store Access' parameters in the Programmer.Set Up area of configuration described in section 4.8.3. Access to some items also depends on whether or not the program is running.

### 3.4.9 PROGRAMMER DISPLAY MODE (cont.)

#### PROGRAM DETAILS

| | |
|---|---|
| Operation | This allows the user to select one of the following (see also 'Program Store, below):<br>Load. Opens the program store and allows the user to select a program to be loaded. The program must have the same number of channels as defined in Programmer.Set Up (section 4.8.3).<br>Store. Allows the current program to be saved to the internal program drive.<br>Delete. Allows the selected program to be deleted.<br>Delete All. Deletes all programs.<br>Copy. Copies the selected program for 'pasting' either from the internal drive to the USB device, or *vice-versa*.<br>Copy All. As above, for 'Copy', but copies all the programs in the selected directory. |

Note: If a 'Store', 'Copy' or 'Copy All' operation would result in there being a total of more than 100 program files in the internal drive, the operation fails and an error message is displayed.

| | |
|---|---|
| Status | Success. Previous operation was successful.<br>Failed. Previous operation failed.<br>Loading. The program is loading.<br>Copying. The program copy process is underway.<br>Deleting. The relevant program is being deleted. |
| Program | The name of the program currently loaded. |
| Holdback Style | Appears only if 'Holdback' is enabled in the Programmer Features configuration (section 4.8.1). See also 'Holdback', below.<br>Program: Holdback applies to all appropriate segments.<br>Per Segment: Holdback enabled on a segment by segment basis as described in 'Segment configuration below. |
| Ch1 Holdback | Appears only if 'Holdback Style' (above) is set to 'Program'.<br>Off: Holdback is disabled<br>Low: Holdback is entered when PV < (PSP - Holdback Value)<br>High: Holdback is entered when PV > (PSP + Holdback Value)<br>Band: Holdback is entered when PV < (PSP - Holdback Value) or PV > (PSP + Holdback Value) |
| Ch1 Holdback value | The value to be used in triggering holdback. |
| Ch2 Holdback | As for Ch1 Holdback, above but for channel 2. Appears only if 'Channels' is set to '2' in Programmer Set Up configuration (section 4.8.3). |
| Ch2 Holdback value | As for 'Ch1 Holdback value', above, but for channel 2. Appears only if 'Channels' is set to '2' in Programmer Set Up configuration (section 4.8.3). |
| Ramp Style | Ramp style applies to all ramp segments in the program. Ramp Style can be edited only when the program is in Reset mode. Setpoints, rates, times etc. are set in the individual segment configurations<br>Rate. A Ramp Rate segment is specified by a target set-point and the rate at which to ascend/descend to that set-point.<br>Time. A Ramp Time segment is specified by a target set-point and a time in which to achieve that set-point. |
| Ch1 Ramp Units | Select 'Per Second', 'Per Minute' or 'Per Hour' for ramp timing units. Ramp Units can be edited only when the program is in Reset mode. |
| Ch2 Ramp Units | As for 'Ch1 Ramp Units' above. Appears only for two channel programs and allows different ramp units to be selected for the two channels, if required. Ramp Units can be edited only when the program is in Reset mode. |

### 3.4.9 PROGRAMMER DISPLAY MODE (cont.)

### PROGRAM DETAILS (Cont.)

HOLDBACK

Holdback pauses the program (freezes the Programmer setpoint (PSP) and the time remaining parameters) if the difference between the Process value (PV) and the PSP exceeds a user-specified amount (Holdback value). The program remains paused until the PV returns to within the specified deviation.

In ramp or step segments, holdback indicates that the PV is lagging the SP by more than the specified amount and that the program is waiting for the process to catch up. In a dwell segment, holdback is used to guarantee that a work piece stays at set-point within a specified tolerance for the specified dwell duration.

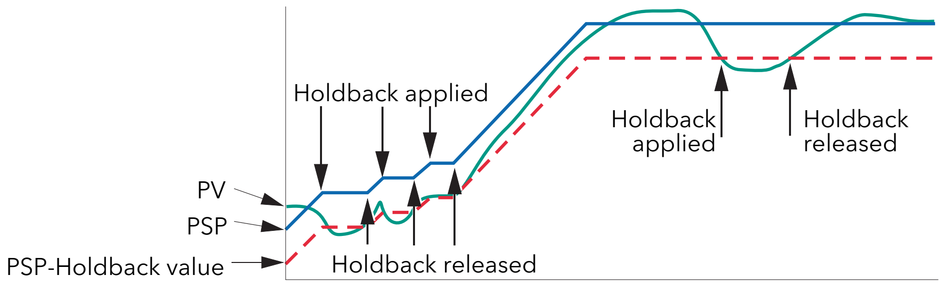



Figure 3.4.9e Holdback

### SEGMENT CONFIGURATION

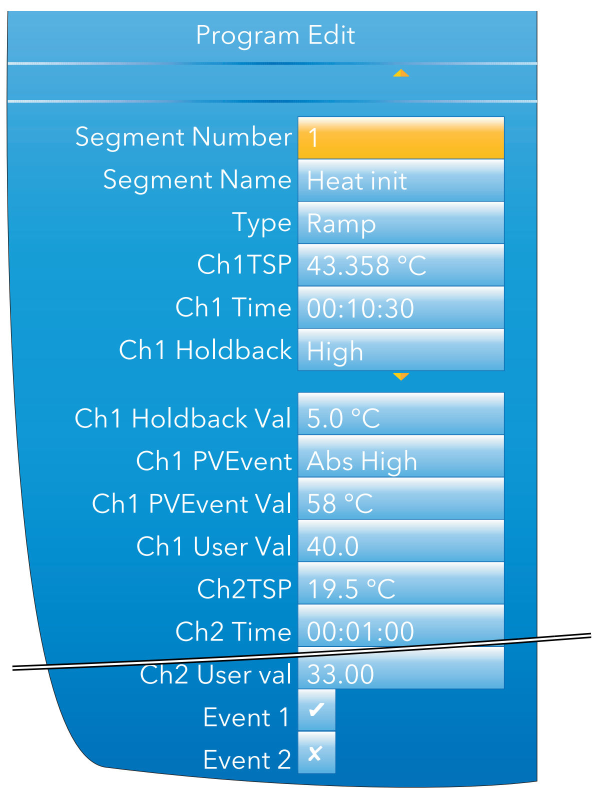



Figure 3.4.9f Segment configuration

| | |
|---|---|
| Segment Number | Select the relevant segment for configuration. |
| Segment Name | Enter a segment name of up to 20 characters. This name will be truncated on the display page if it, together with the program name, are too long to fit the width of the display area. |

### 3.4.9 PROGRAMMER DISPLAY MODE (cont.)

**SEGMENT CONFIGURATION (Cont.)**

| | |
|---|---|
| Type | Select a segment type. Default is 'End'.<br>Ramp. For any program, Ramp segments can be either 'Ramp Rate' segments or 'Ramp Time' segments according to the 'Ramp Style' setting described above. See also 'Ch1(2) Time' or 'Ch1(2) Rate', below.<br>Dwell. The setpoint is maintained at its current value for the period defined in 'Duration' (see below).<br>Step. A step segment allows a step change to be entered for the target set-points Ch1 TSP and Ch2 TSP.<br>Wait. A wait segment causes the program to wait for a certain event to occur before continuing. See 'Wait For', below.<br>Go Back. A Go Back segment allows a specifiable number of iterations to be performed of a group of segments. This could be used, for example, to cycle an entire program by having a Go Back segment immediately before the end segment and specifying segment 1 as the 'Go Back To' point. Setting 'Cycles' to 'Continuous' causes the program to loop indefinitely, until interrupted by the user. 'Nested' loops are not permitted i.e. 'Go Back' is not available as a segment type for segments inside an existing GoBack loop.<br>End. The final segment of a program allows the user to select 'Dwell' or 'Reset' as the action to be taken at the end of the program (see 'End Type', below) |
| Ch1(2) TSP | Target setpoint. The value that Ramp or Step segments seek to attain, for channel 1(2). |
| Ch1(2) Rate | For Ramp Rate segments, this specifies the speed at which the process value ramps towards the target, for Channel 1(2). The ramp units (per second, per minute, per hour) are set in Ch1(2) ramp units described above. |
| Ch1(2) Time | For Ramp Time segments, this allows the user to specify the time to be taken by the segment for the process value to reach the target. |
| Duration | For Dwell segments, this allows the entry of the time for which the segment dwells. |
| Go Back To | For 'Go Back' segments, this defines the number of the segment to which the program is to return. |
| Cycles | The number of times the 'Go Back' instruction is to be carried out. If set to 'Continuous', the program continues until the user intervenes to stop it. |
| End Type | Allows the user to select the action to be taken at the end of the program:<br>Dwell: the set-point is maintained indefinitely and event outputs remain at their configured state.<br>Reset: the set-point reverts to the value used by the control loop before the program was started and the event outputs return to their default states. |
| Wait For | Digital High: Wait segments can be configured to wait for 'Wait Digital' to go 'high' before allowing the program to continue.<br>Analog 1(2): The segment waits for 'Wait Analog1(2) to meet an Absolute High or Low, or Deviation High or Low condition before allowing the program to continue.<br>Analog Both: As Analog 1(2) above, but waits for both Channels' conditions to be true before continuing. |

Note: 'Wait Digital', Wait Analog 1' and 'Wait Analog 2' parameters are configured in the Programmer.Set Up menu described in section 4.8.3.

| | |
|---|---|
| Ch1 Wait | Select 'Abs High', 'Abs Low', 'Dev High' or 'Dev Low' as the wait criterion for channel 1. Appears only if 'Wait For' (above) is set to 'Analog 1' or 'Analog Both'. |
| Ch2 Wait | Select 'Abs High', 'Abs Low', 'Dev High' or 'Dev Low' as the wait criterion for channel 2. Appears only if 'Wait For' (above) is set to 'Analog 2' or 'Analog Both'. |
| Ch1(2) Wait Val | Enter the trigger value for 'Ch1(2) Wait' |
| Ch1(2) Holdback | Select 'Off', 'Low', High', or 'Band' (see description in Program details above). |
| Ch1(2) Holdback Val | The value to be used in triggering holdback. |

### 3.4.9 PROGRAMMER DISPLAY MODE (cont.)

**SEGMENT CONFIGURATION (Cont.)**

Ch1(2) PV Event — Appear only if 'PV Events' have been enabled in the Programmer Features menu (section 4.8.1). A PV Event (an analogue alarm on the channel PV) is available for each channel in every segment (excluding Wait and Go Back segment types). The following PV Events are supported:
Off: The PV Event is disabled
Abs High: The event is triggered when the channel PV exceeds PVEvent Val for the relevant channel.
Abs Low: Triggered when the channel PV becomes less than PVEvent Val for the relevant channel.
Dev High: This event is triggered when the channel PV exceeds (PSP + PVEvent Val) for the relevant channel
Dev Low: Triggered when the channel PV becomes less than (PSP - PVEvent Val) for the relevant channel.
Dev Band - This event is triggered when the channel PV differs from the PSP by more than the configured deviation value (either above or below)
In the following example, in segment 1 Ch1 PV Event has been configured as Dev Band and in segment 2 it has been configured as an Abs low:

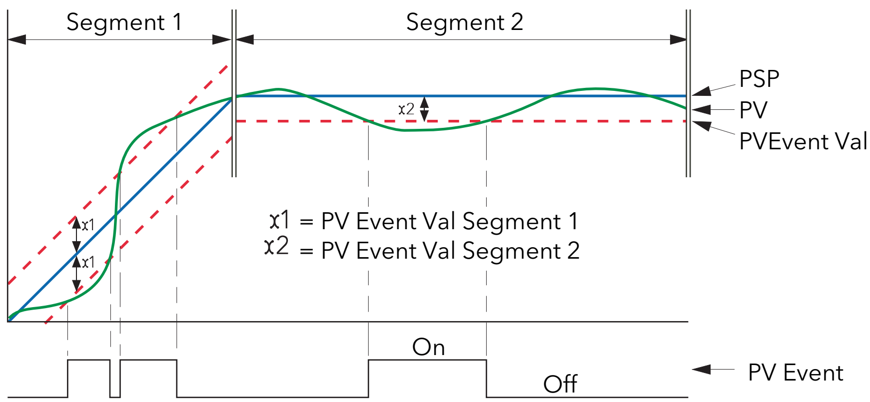



Figure 3.4.9g PV Events

Ch1 PVEvent Val — Appears only if 'Ch1 PVEvent' is not 'Off'. Sets the level at which Ch1 PV Event becomes active.

Ch2 PVEvent Val — Appears only if 'Ch2 PVEvent' is not 'Off' and if 'Channels' is set to '2' in Programmer Set Up configuration (section 4.8.3). Sets the level at which Ch2 PV Event becomes active.

Ch1 (2) Event Use — When PV events become active, they can be used either to Trigger a secondary process or as a simple analogue alarm on the PV input. Appears only if the relevant PV Event parameter is not set to 'Off'.

Ch1 (2) User Val — Specifies the User Value for this segment, for channel 1(2). Appears only if 'User Value' has been enabled in the Programmer Features menu (section 4.8.1).
The example below (from iTools) shows this parameter wired to the trigger 1 input of the Custom Messages block, so that, if a User value >0 is entered, then every time the segment runs, Custom message 1 is generated.

Event 1 to 8 — The number of Events available (Max Events) is defined in Programmer Set Up configuration (section 4.8.3). Enabling an event causes the relevant indicator on the display page to be illuminated for the duration of the segment. As with 'User Val', above, Events can be wired to the inputs of other parameters if required.

### 3.4.9 PROGRAMMER DISPLAY MODE (cont.)

### SEGMENT CONFIGURATION (Cont.)

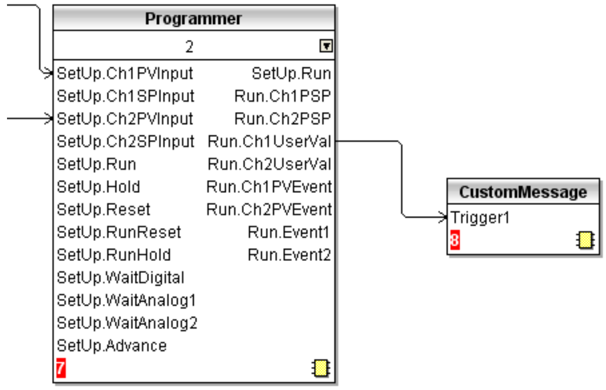



Figure 3.4.9h iTools example showing Ch1 UserVal being used to trigger custom message 1.

### FUTURE TREND DISPLAY MODE

If enabled in Instrument.Display configuration (section 4.1.3), this allows the user to view the actual value of the PSP alongside the expected value, so the two can be compared to see how the process is performing.

Future trend is an enhancement of the horizontal trend mode, with the display being divided into two parts, with the instantaneous current value located at the divide, with past trends to the left and the next few program segments to come, to the right.

Notes:

1. For the future trend mode to appear, the programmer must be wired to the loop or advanced loop feature.
2. Both historic and future trends move from right to left with the present anchored at the screen centre.
3. The amount of history and of future trending displayed on the screen depends on the trend interval set in Group.Trend configuration (section 4.3.1)

Figure 3.4.9i shows a typical future trend display

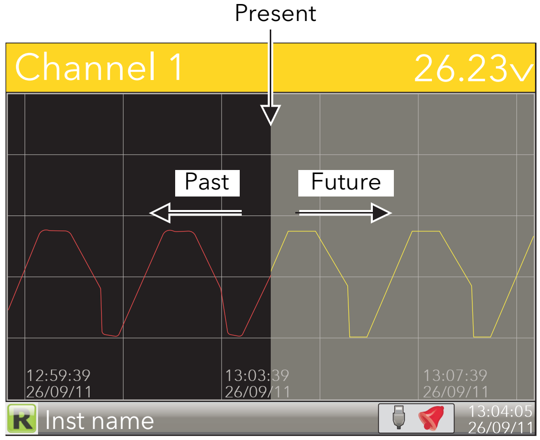



Figure 3.4.9i Future trend display

### 3.4.9 PROGRAMMER DISPLAY MODE (cont.)

#### PROGRAM STORE

Note: The access levels required for the operations described below are configured in the Programmer Set Up menu 'Prog Edit Access' and 'Prog Store Access' parameters, described in section 4.8.3.

The program store allows access to the instrument's local program storage area and to programs stored on a USB memory stick (if any) and to those stored in a pc (if any), via FTP. Programs may be saved to (Stored) or retrieved from (Loaded) from the program store, or they can be copied or deleted.

Selecting any of the program operations (except 'Delete All'), from the Program Edit page (Engineer access level required) opens the file explorer page. Figure 3.4.9j depicts this page, with just a couple of example entries after a 'Load' operation has been requested.

On entry, use the up/down arrow button to select 'User', 'USB' or 'FTP' (selection highlights yellow), then use the scroll button to confirm. Use the up/down arrow buttons to select the required file, and then use the scroll button again to confirm. Other operations are similar.

The file explorer supports 100 entries, which may be directories or files.

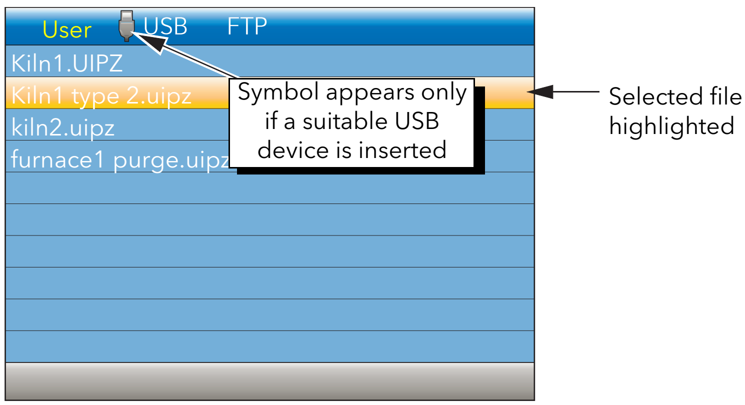
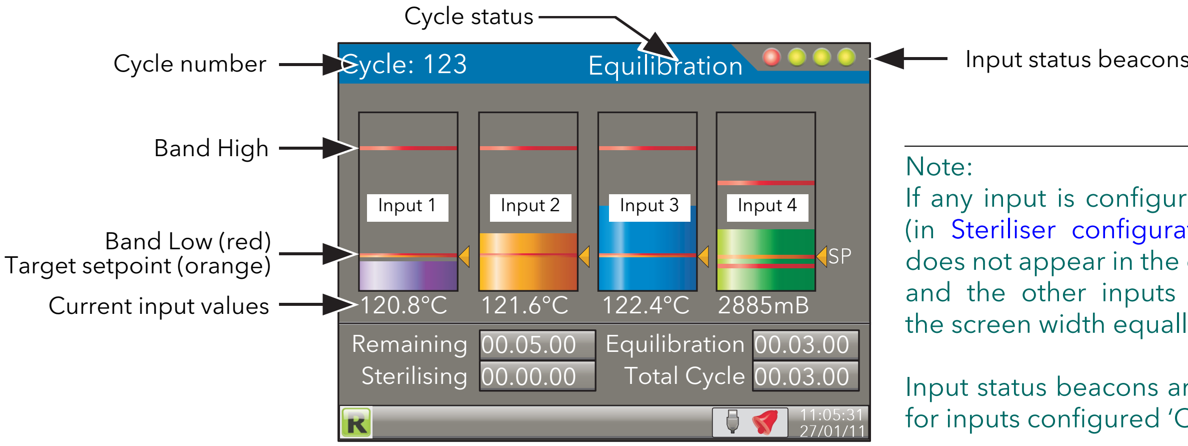



Figure 3.4.9j Program store display

Note: A 'busy' icon (rotating green flash) appears whilst directory listings are being accessed.

### 3.4.10 Steriliser display mode

This display mode appears only if the Steriliser option is fitted and if the display mode has been enabled in the Instrument Display configuration (section 4.1.3) Steriliser configuration parameters are to be found in section 4.16.




Note:
If any input is configured 'Off' (in Steriliser configuration), it does not appear in the display, and the other inputs occupy the screen width equally.

Input status beacons are black for inputs configured 'Off'

Figure 3.4.10a Steriliser display mode (typical) (four inputs)

**OPERATION**

A sterilising cycle cannot be initiated whilst the unit is in Configuration (Engineer) mode.

A steriliser cycle is started by setting its relevant 'Start' input to 'Yes' for the duration of the cycle. The cycle waits (status 'Waiting') until input 1 reaches its setpoint, at which point the cycle enters the equilibration period (status 'Equilibration'), and remains there until all the configured inputs are valid. The cycle then enters the sterilising period and stays in this mode until the sterilising period has expired (status 'Passed') or until one of the inputs becomes invalid (status 'Failed') for longer than its configured 'Failure Dwell' time.

Note: The cycle stops (status 'Failed') if the trigger source is removed.

**TERMINOLOGY**

| | |
|---|---|
| Holding time | Most operating cycles have a stage in which the load must be exposed to sterilisation conditions for a specified length of time, known as the 'Holding time'. |
| Equilibration time | The holding time (above) is preceded by a period during which, although the sterilising condition is present in the chamber, the load has not yet attained that temperature due to its thermal inertia. 'Equilibration time' is defined as the time between the attainment of sterilisation temperature in the chamber, and the attainment of that temperature in all parts of the load. |
| Bands | For steam and dry heat sterilisers, sterilisation conditions are specified by a sterilisation temperature band, defined by a minimum acceptable temperature (known as the sterilisation temperature) and a maximum allowable temperature. A sterilisation band is normally quoted for each steriliser type. |

**BEACONS**

There are four input status beacons near the top right hand corner of the display, one for each input.

During equilibration, the beacons are flashing red for inputs that have not attained the Target setpoint, and go green when the target setpoint is reached, remaining green even if the input value rises above the Band High value The beacons revert to red if input falls below* the target setpoint.

During sterilisation, the beacons go red for any input whose value rises above Band High or falls below* setpoint for a duration exceeding the configured 'Failure Dwell' period.

Beacons are black for inputs that are configured as 'Off'.

* 'rises above' for input types 'Falling Pressure' or 'Fall Air Detect'

### 3.4.10 STERILISER DISPLAY MODE (Cont.)

#### DISPLAYED INFORMATION

| | |
|---|---|
| Cycle | A five-digit counter to indicate the total number of cycles started. |
| Status | Wait start: The initial state at power up. This status remains until the first cycle is initiated<br>Waiting: Waiting for input 1 to reach its target setpoint. The cycle then enters Equilibration.<br>Equilibration: Currently in the equilibration period, during which the cycle waits until all inputs have reached sterilisation conditions.<br>Sterilising: Currently in the decontamination phase<br>Passed: The cycle has completed successfully<br>Failed: The cycle has failed either through one or more inputs becoming invalid, or because the 'Start' signal was removed.<br>Test cycle: A test cycle is in progress |
| Remaining | The sterilising time remaining for the current cycle. Display field is replaced by 'Target Time' (below) when the cycle is not running. |
| Target time | The intended sterilisation time. This can be configured by operating the scroll button twice (once to highlight the field, and again to enter edit mode), and then using the up and/or down arrows to edit the time. Use the Scroll button again to quit edit mode, and the page key to 'unhighlight' the field.<br>Replaced by 'Remaining' (above) when the cycle is running. |
| Equilibration | The equilibration time period for the current cycle |
| Sterilising | The time for which the load has currently been at sterilisation conditions |
| Total Cycle | The elapsed time since the initiation of the current cycle. This time increments from the time the cycle is triggered until the time the trigger is removed. |
| Input values | Temperature are required in °C; pressure inputs in mBar. If necessary, maths channels and user values can be used to convert from other units (see 'Note' overleaf). |

#### STERILISING CYCLE DIAGRAM

Figure 3.4.10b, below, shows a steriliser cycle in diagrammatic form.

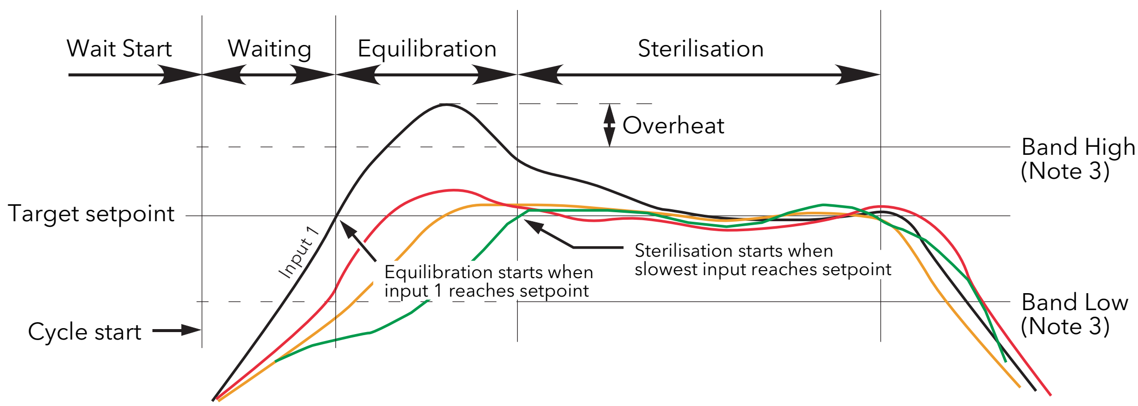



Figure 3.4.10b Steriliser cycle

Notes:

1. For temperature inputs in most applications, the Setpoint value is the same as the Band Low value For the sake of clarity, this is not as shown in the figure above.
2. For the sake of clarity all four inputs in the figure above are shown with the same Band High, Band Low and Setpoint value. This would not be unusual for temperature units, but the pressure input would normally have a different set of values from temperature inputs.
3. Band High and Band Low are effective only during Sterilisation phase.

### 3.4.10 STERILISER DISPLAY MODE (Cont.)

#### APPLICATION DETAILS

Figure 3.4.10c shows a typical steriliser application, with temperature and pressure signals from the sterilisation chamber being applied directly to the rear terminals of the controller/recorder, and control signals connected from the controller to both the chamber and the controller/recorder.

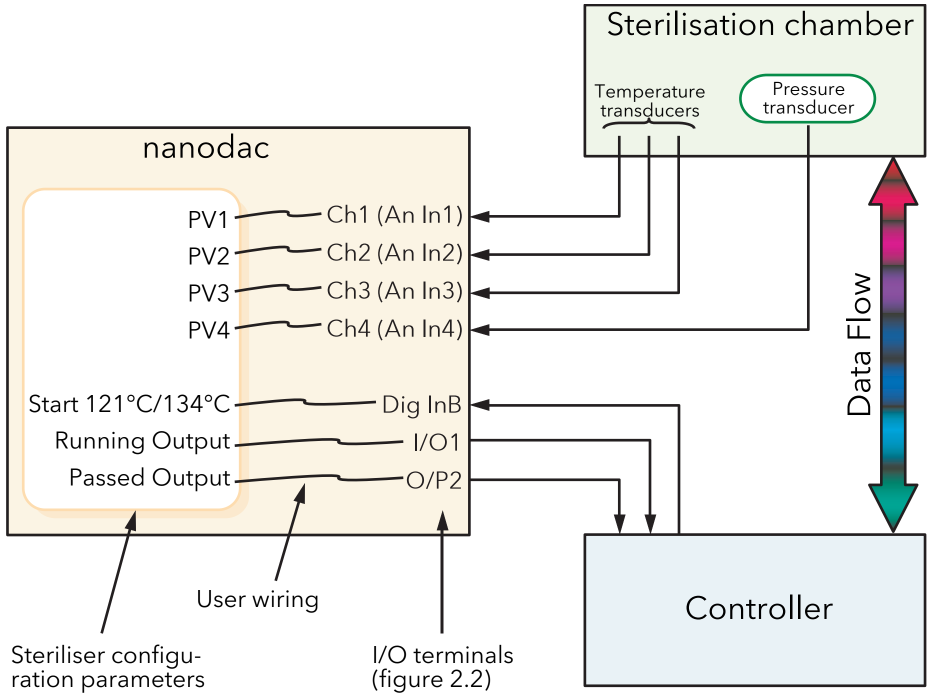



Figure 3.4.10c Typical steriliser application

Analogue inputs 1 to 3 receive signals from temperature transducers (typically thermocouples) within the chamber. These inputs are internally connected to channels 1 to 3 respectively, allowing transducer type, ranges, alarms etc. to be configured (section 4.4). Inputs are assumed to be degrees Celsius*.

The pressure transducer is connected to channel 4 and can be configured in the same way. The input is assumed to be in milliBar. Other pressure inputs should be converted using virtual channels*.

PV1 to PV4 in the Steriliser configuration is software wired (section 7) to Ch 1 to Ch4.

Start cycle input and the 'Running Output' and 'Passed Output' signals are software wired to suitable DIO terminals, for connection to the Controller.

*Note: For Fahrenheit inputs, use one virtual channel to subtract 32, and a second to divide the result by 1.8 (where 32 and 1.8 can be configured as user values). Similar techniques should be used to convert pressure input units if necessary.

#### TEST CYCLES

A 'Test' cycle is initiated by initiating a 121°C cycle and a 134°C cycle simultaneously. A test cycle allows the user to check actual performance against expected performance.

### 3.4.10 STERILISER DISPLAY MODE (Cont.)

**$F_0$**

$F_0$ is a means of calculating 'equivalent time at sterilising temperature' for temperatures below, at and above sterilizing temperature, using the equation below.

$$F_0 = \text{Sterilisation time} \times 10^{\frac{Temp - Ts}{Z}}$$

Where:

| | |
|---|---|
| Sterilisation time | Depends on the application, typically 15 minutes at Ts = 121°C |
| Temp | The value of the temperature measuring input. |
| Ts | Desired Sterilising temperature |
| Z | Temperature interval representing a factor-of-10 reduction in killing efficiency. Z = 10 for steam sterilising ($F_0$), or Z=20 for dry heat sterilising (FH). Z = 10 for thermal disinfection ($A_0$). |

To ensure that steriliser loads which contain materials with different thermal inertias are thoroughly sterilised, a number of sensors are located withing the load. The F value should be calculated using the sensor closest to that part of the load which has the highest thermal inertia. For maximum accuracy, the temperature sensor should be calibrated and the input adjust function used to compensate for any inaccuracy found.

F0 calculation examples

For all the examples following, the following are assumed: Sterilisation time = 15 minutes; Sterilisation target temperature =121°C and Z = 10.

1. For an actual sterilising temperature of 111°C

$$Fval = 15 \times 10^{\frac{111-121}{10}} = 15 \times 10^{\frac{-10}{10}} = 1.5 \text{ minutes}$$

Which means that 15 minutes at 111°C is equivalent to 1.5 minutes at 121°C

2. For a sterilising temperature of 121°C

$$Fval = 15 \times 10^{\frac{121-121}{10}} = 15 \times 10^{\frac{0}{10}} = 15 \text{ minutes}$$

Which means that the sterilising temperature is ideal (by definition)

3. For a sterilising temperature of 124°C

$$Fval = 15 \times 10^{\frac{124-121}{10}} = 15 \times 10^{\frac{3}{10}} = 15 \times 1.995 = 29.925 \text{ minutes}$$

Which means that 15 minutes at 124°C is equivalent to nearly 30 minutes at 121°C.

Normally sterilising temperatures would not remain constant at temperatures below or above the target value, so the above equations are illustrative only of the facts:

1 Temperatures below the target have some killing efficacy

2 Temperatures above the target value have a greater killing efficiency, so that the sterilising time can be reduced.

In order to calculate the value dynamically, the instrument uses the equation:

$$Fval_t = Fval_{t-1} + T \times 10^{\frac{ma_t - \text{Target temp}}{Z}}$$

where

| | |
|---|---|
| $Fval_t$ | = F value this iteration |
| $Fval_{t-1}$ | = F value last time |
| T | = Iteration period (minutes) |
| $ma_t$ | = input temperature value this iteration |
| Target Temp | = 121°C for $F_0$, 170°C for $F_H$, 80°C for $A_0$ |
| Z | = 10°C for $F_0$, 20C for $F_H$, 10°C for $A_0$ |

### 3.4.11 Promote list

This display page allows the user to display up to 10 of the parameters that appear anywhere in the operator interface. The parameters can be selected only by using iTools, as described below.

Notes:

1. 'Promote List' must be enabled (in 'Instrument.Display' configuration), before it appears in the 'Go to View' list.
2. There are more parameters visible in iTools than appear at the operator interface. If non-operator interface parameters are selected for inclusion in the promote list, they do not appear.
3. If parameters which appear only in certain circumstances are selected, then they appear in the promote list only when they appear in the Operator interface. For example, a channel PV is not visible unless that channel is enabled (i.e. it is not 'Off').

**PARAMETER SELECTION**

1. Open iTools and scan for the instrument, (see section 6).
2. Once the instrument has been found, stop the scan. When the instrument has synchronised, click on the 'Access' button near the top of the display to set the unit into configuration mode (a password may be required).
3. Click on the '+' sign to the left of the Instrument folder in the tree list (left-most pane) to expand the folder. Double-click on 'Promote List', to display the Promote list in the main pane. The list contains 20 entries, 1 to 10 being for parameters, 11 to 20 being available to the user to add descriptors for parameters 1 to 10 respectively.
4. Expand further folders, as necessary, to access the required parameters, and click-drag these parameters into the promote list. Enter a descriptor for the parameter if the default is not as required. As each parameter is dragged into the list, it appears in the Promote list.
5. If the parameters are modified at the operator interface, the changes are reflected in iTools, and *vice-versa*.
6. Once all the parameters have been added, it is recommended that the Access button be used to quit configuration mode, as otherwise it will not subsequently be possible to quit from the operator interface.

Figure 3.4.11 shows typical displays.

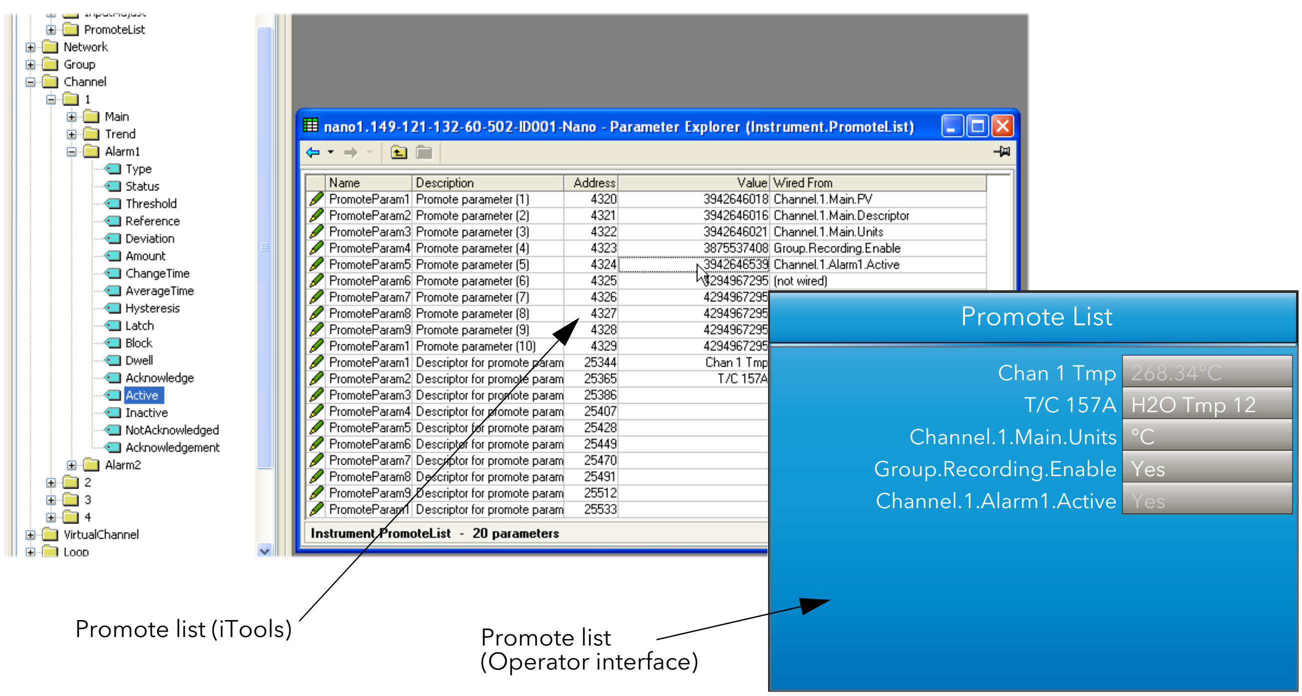
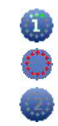



Figure 3.4.11 Promote list displays.

### 3.4.12 Modbus Master display mode

This display mode consists of two pages, as shown below.

Page one opens by default and shows the first eight parameters being read from (left pointing arrow) or written to (right pointing arrow) the relevant slave. These items are configured in the Modbus Master configuration described in section 4.9. Hidden parameters may be viewed by operating the scroll key, then using the arrow keys to scroll through the list. A green arrow means that the item may be edited by the user when logged in.

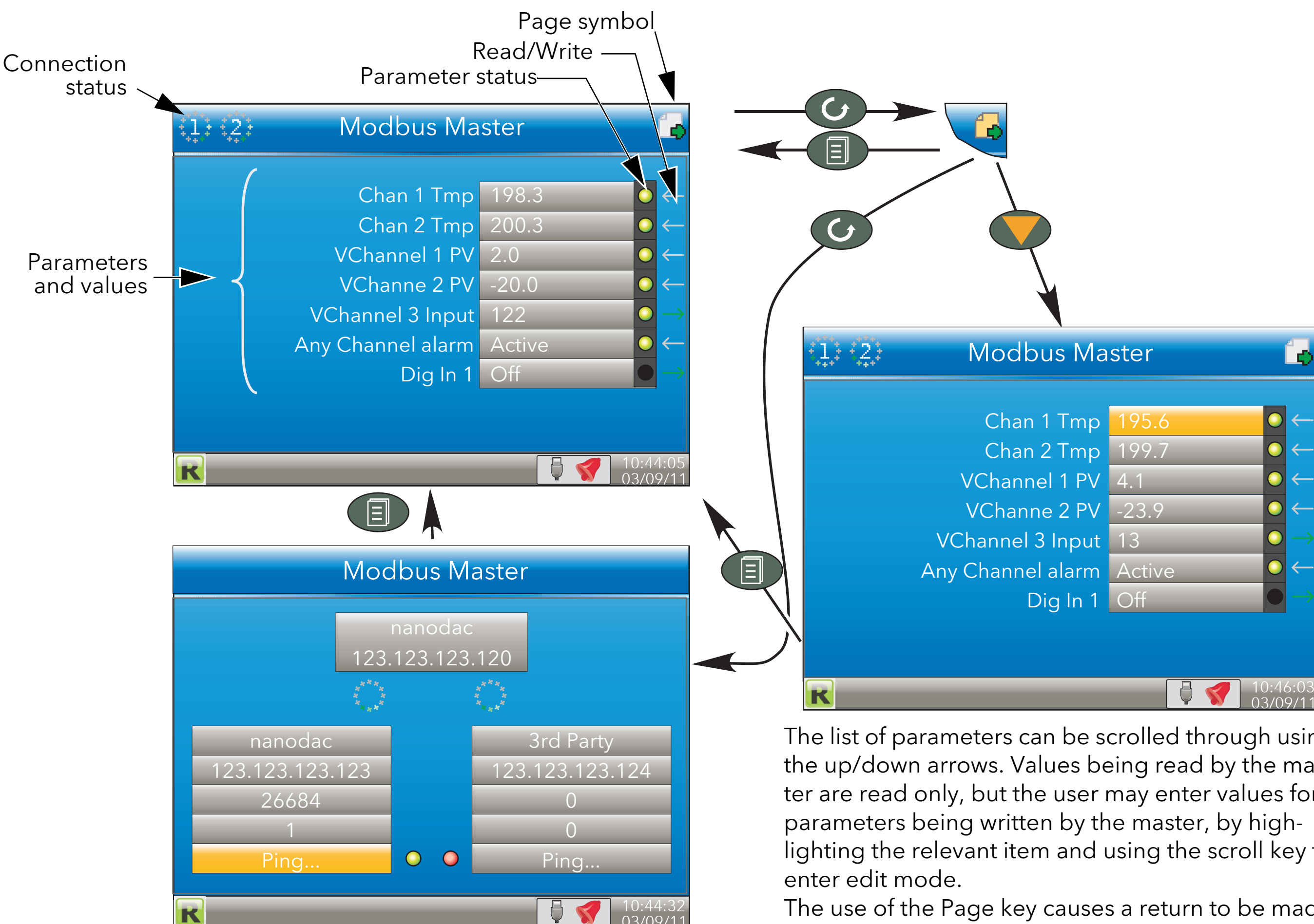


A pair of animated indicators in the top left-hand corner of the screen show the connection status of the two possible slaves. A green moving 'streak' indicates that successful communications are being carried out. A red flashing circle indicates that there is a break in the transmission line or that the slave is switched off. A grey, non-animated display indicates that that slave has not yet been configured as a part of the communications link (i.e it is 'off line').

A 'traffic light' indicator appears to the right of each parameter. Green indicates that the parameter is being read from or written to successfully. Orange indicates that a write of the value is pending. Red indicates that there is an error and that no value is currently being read or written; the value displayed is the last good value read or written depending on whether the data item is a read or write. If the indicator is black, the parameter is 'off'.

Operation of the scroll key highlights the page symbol in the top right-hand corner of the screen, and a further operation of the scroll key calls page two to the screen.

Page two contains the IP address of the Modbus master and of any slaves connected to it, together with some diagnostic information, as described in 'PING DETAILS' below.




The list of parameters can be scrolled through using the up/down arrows. Values being read by the master are read only, but the user may enter values for parameters being written by the master, by highlighting the relevant item and using the scroll key to enter edit mode.

The use of the Page key causes a return to be made to the initial Modbus Master display.

Figure 3.4.12a Modbus Master display pages

### 3.4.12 MODBUS MASTER DISPLAY MODE (Cont.)

#### PING DETAILS

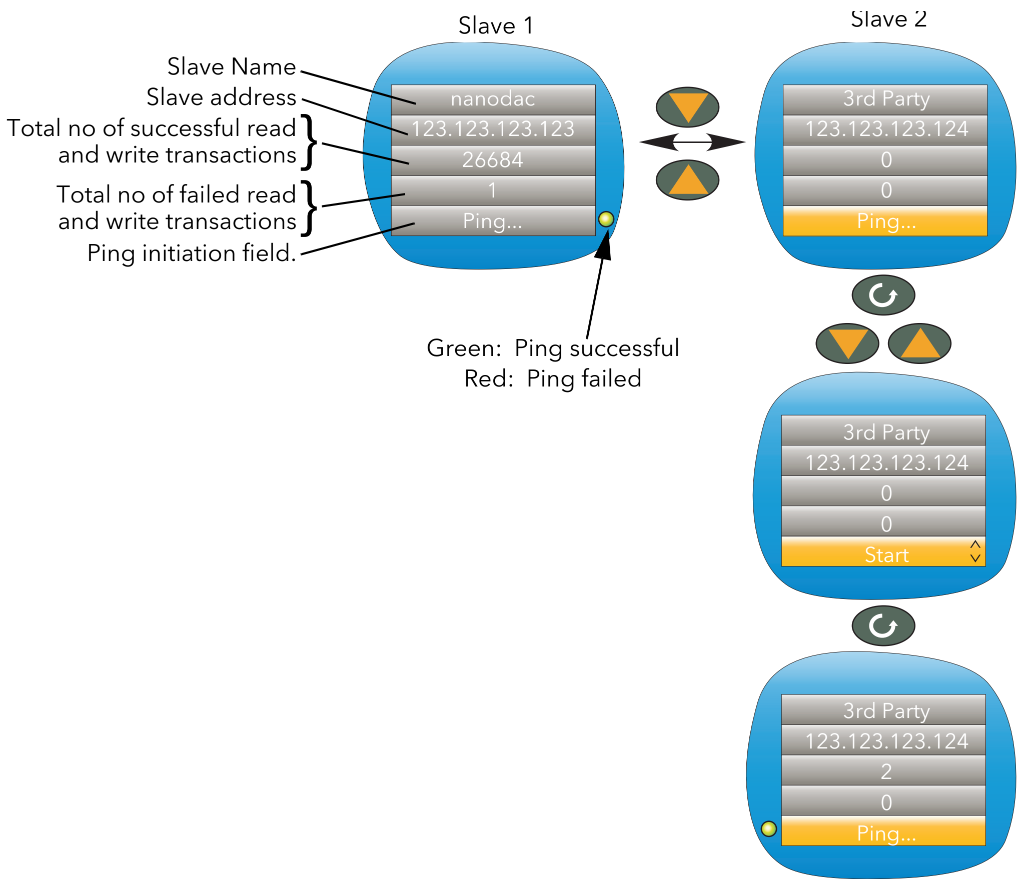



Figure 3.4.12b Slave 2 ping initiation (Slave 1 similar)

The 'Ping...' field of the first slave is highlighted by default. As shown above, the down (or up) arrow can be used to highlight the 'Ping...' field of the other slave instead.

Once the relevant 'Ping...' field is highlighted, the scroll key can be used to enter edit mode and the up/down arrow key used to select 'Start'. A further operation of the scroll key initiates the 'Ping' and if this is successful, a green indicator appears alongside the field (and the text returns to 'Ping...'). If the Ping is unsuccessful, then the indicator is coloured red.

The up or down arrow can now be used to return to slave 1, or the page key can be used to return to the previous parameter display page.

As shown in the figure above, some diagnostic information is given. This includes the total number of successful attempts that the master has made to communicate with the relevant slave, and the total number of failed attempts. Fuller diagnostic details are to be found in the Modbus Master Communications configuration description (section 4.9)

### 3.4.13 EtherNet/IP display mode

This display mode appears only if enabled in Instrument.Display configuration ( Section 4.1.3) and is used to display the input and output parameters assigned to the Client and Server input and output tables. Parameters which have been configured with descriptors are identified by these descriptors instead of their ‘opc’ names (shown below).

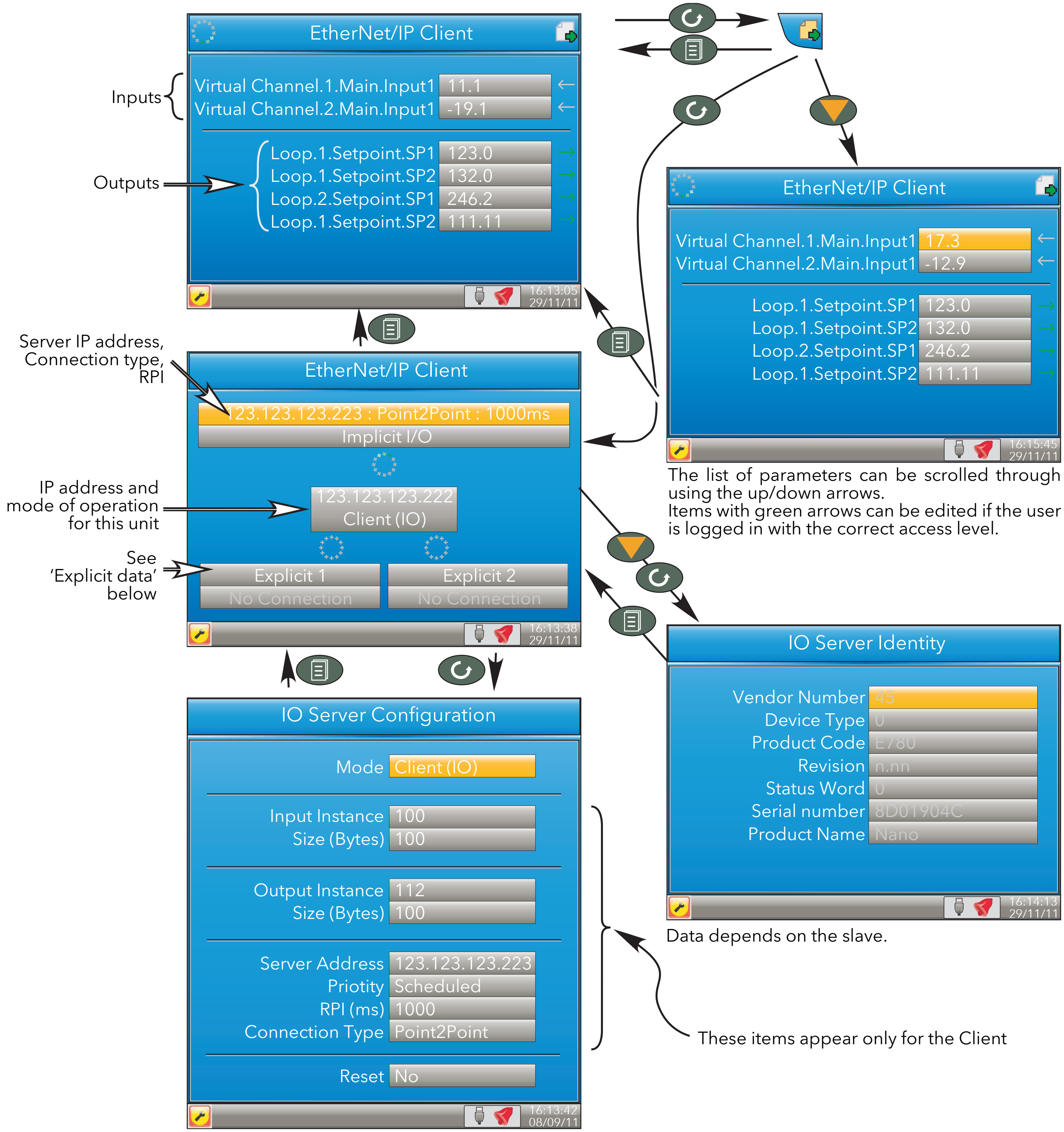



Figure 3.4.13a Typical EtherNet/IP display

### 3.4.13 ETHERNET/IP DISPLAY MODE (Cont.)

If the EtherNet/IP option is fitted, the nanodac can be configured as either a client (master) or a server (slave) (see section 4.10). The client and server displays are identical except that the configuration area of the client display is more extensive than that of the server display.

Figure 3.4.13a, above shows a typical set of display pages for an EtherNet/IP client.

### CONFIGURATION OF IMPLICIT INPUT/OUTPUT TABLES

Configuration of the input and output tables is carried out by:

a. Entering the parameters to be read by the client into the server output table.

b Entering the destination parameter into the equivalent location in the client input table.

c. Entering the parameters to be written by the client into the client output table.

d Entering the destination parameter into the equivalent location in the server input table.

The example in figure 3.4.13b attempts to show this (using the nanodac as the client) in graphical form, using just a few parameters (there can be up to 50 in each table).

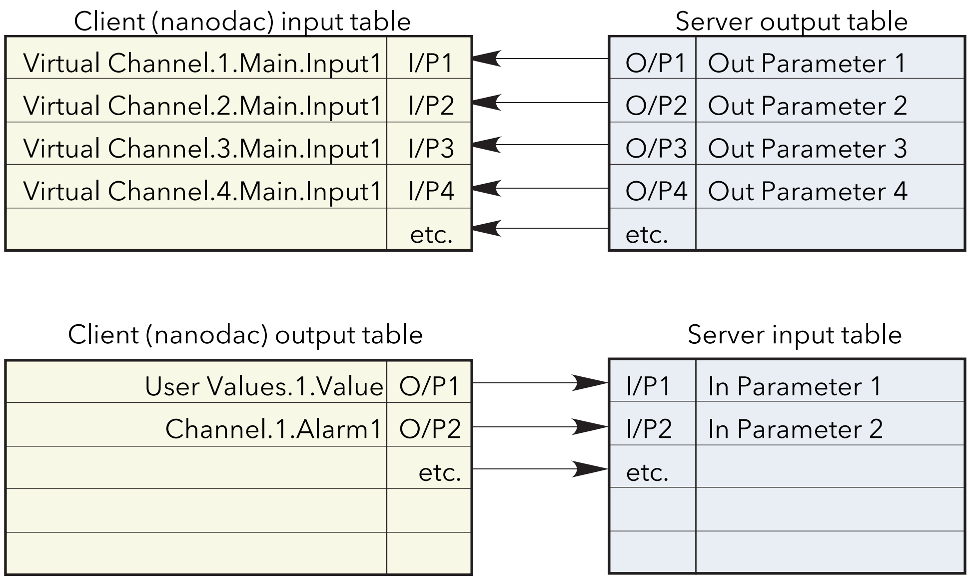



Figure 3.4.13b Input/Output table entries

Notes:

1. Channel values from the Server can be 'wired' into nanodac Virtual channel inputs (as shown above) so that they can be traced and/or recorded. In such cases the virtual channel 'Operation' must be set to 'Copy' (see section 4.5.1).
2. Inputs and outputs would normally be given suitable descriptors (e.g. 'Reset timer' instead of 'Channel.1.Alarm1').

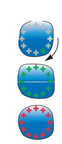
### CONNECTION STATUS INDICATOR

A circular status indicator appears in a number of the EtherNet/IP display pages. This indicator can indicate the following states:

Green rotating 'flash': the instrument is on line and at least one CIP connection is established.

Green flashing circle: the instrument is on line but no CIP connections have been established.

Red flashing circle: there is a break in the physical connection between the client and the server, or the remote unit is switched off or is initialising.

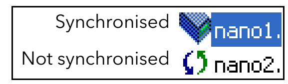


### 3.4.13 ETHERNET/IP DISPLAY MODE (Cont.)

Adding parameters to the input and output tables can be achieved only through the proprietary software package 'iTools', running on a pc. The following description assumes that the user is familiar with 'iTools'. Section 6 of this manual shows how to set up an iTools link to the unit and the iTools on-line help system and its pdf version (HA028838) should be referred-to as necessary.

Note: the client/server and the pc must all be on the same network.

Once iTools has started up and the 'Scan' process has 'found' the relevant instrument, the scan process should be stopped and the instrument (s) allowed to synchronise. (The scan may be left to run its course, but the speed at which iTools operates is reduced for the duration of the scan process.)



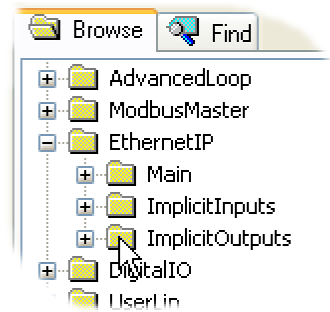

**EXAMPLE**

To add Loop 2 Setpoint 2 to Output 4 of the Client Output table.

In the example shown below, the instruments have both synchronised, and the 'Access' tool button clicked-on for both instruments to set them into configuration mode.

With the client selected, expand the EtherNet/IP folder in the Browse list, then double-click on the 'ImplicitOutputs' folder.

Locate and expand the Loop 2 SP folder in the Browse window, and click-drag SP2 to 'Output 4' and release.



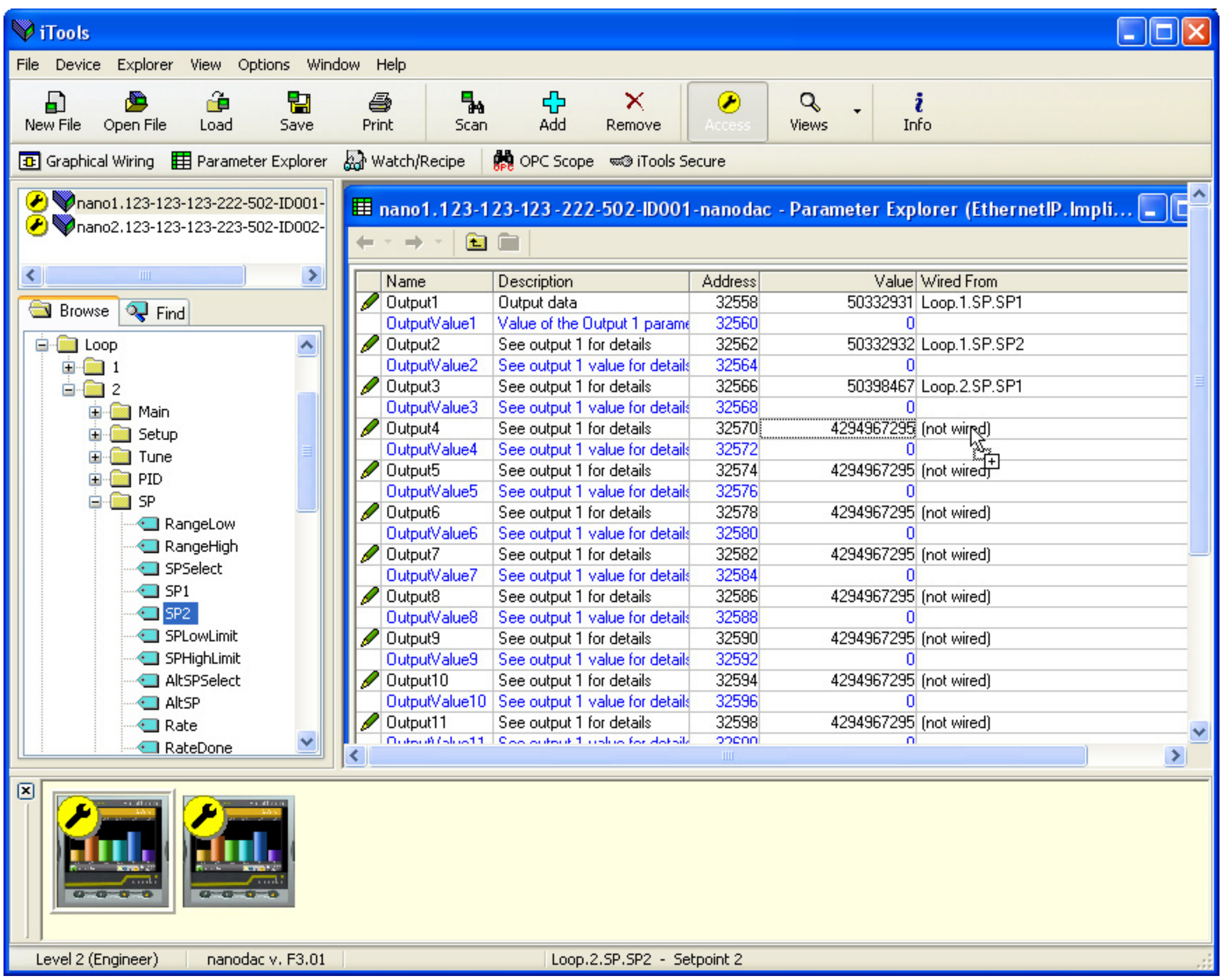




Figure 3.14.3c Dragging a parameter to the Output table

### 3.4.13 ETHERNET/IP DISPLAY MODE (Cont.)

An alternative to the click-drag technique is to right click on the required output (five in the example below), and select 'Edit Wire...' from the context menu that appears. A browse window pops up, allowing the user to navigate to the required parameter. This technique can be used both on previously empty inputs or outputs and on those previously filled.

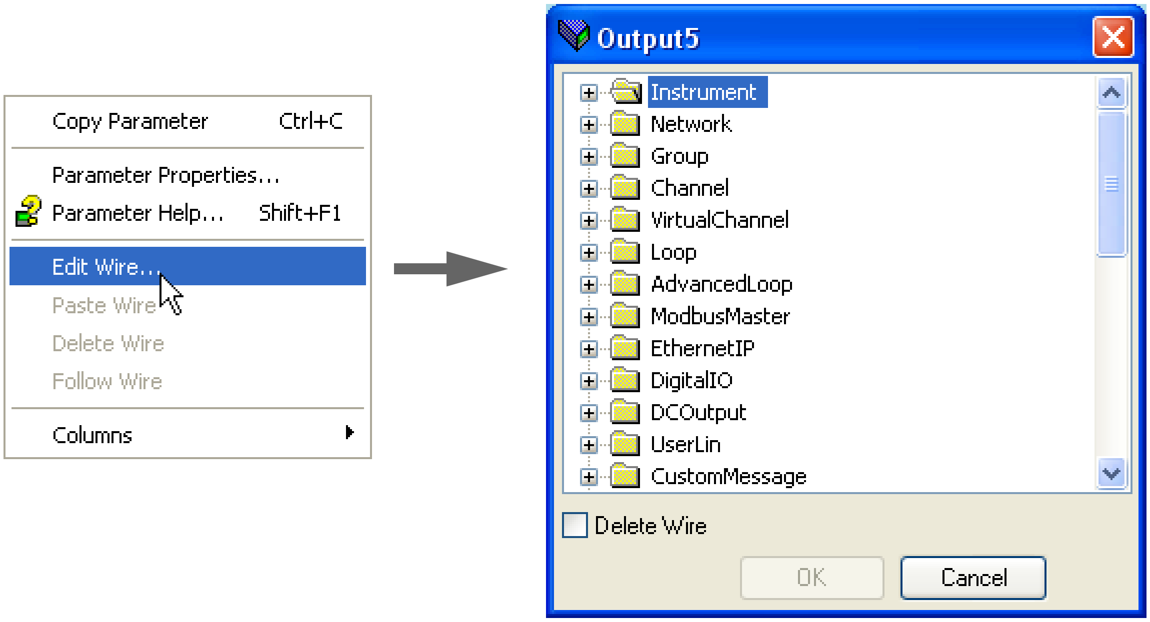



Figure 3.4.13d Context menu details

### EXPLICIT DATA

As shown in table 3.4.13, when configured as a server, there is only one explicit application object, and that has the class ID= A2 (162 decimal). The instance ID is the Modbus address of the parameter and the Attribute is always = 1. Explicit service codes hex10 (decimal 16) and 0E (14) are both supported, for writing and reading single attributes respectively.

| Service code | | Class ID | | Instance ID | Attribute |
|---|---|---|---|---|---|
| Hex | Dec | Hex | Dec | Decimal | |
| 0010 | 16 | A2 | 162 | 1-65535 | 1 |
| 000E | 14 | A2 | 162 | 1-65535 | 1 |

Table 3.4.13 Explicit data specification

When configured as a client, two separate connections are available allowing the user to produce two independent explicit read or write messages to different server devices.

Figure 3.4.13e below, shows an example of how to configure an explicit message request. The instance ID and the data type are taken from the server manufacturer's data. In this example a read request is configured to determine the Group recording status of a nanodac server, and it can be seen from table 5.3 that the decimal modbus address for this parameter is 4150 and the data type is int16. It is this address which is used as the instance ID.

Once all the information has been entered, the read is requested by setting 'Send' to 'Yes'. The Data field changes to '3' for this example and from table 5.3 it can be seen that the recording status is 'Recording enabled'.

Note: The nanodac supports only 16 bit data types for reading and writing of explicit messages.

### 3.4.13 ETHERNET/IP DISPLAY MODE (Cont.)

When wired to parameters which have descriptors, the descriptor names appear instead of the 'opc' names (e.g. Loop.1.Setpoint.SP1) shown here.

Tags page appears if 'Mode' is set to 'Client (Tags)' in EtherNet/IP configuration (section 4.10). Explicit messaging is not possible in tag IO mode as tag mode needs to consume both connections for communications with the PLC.

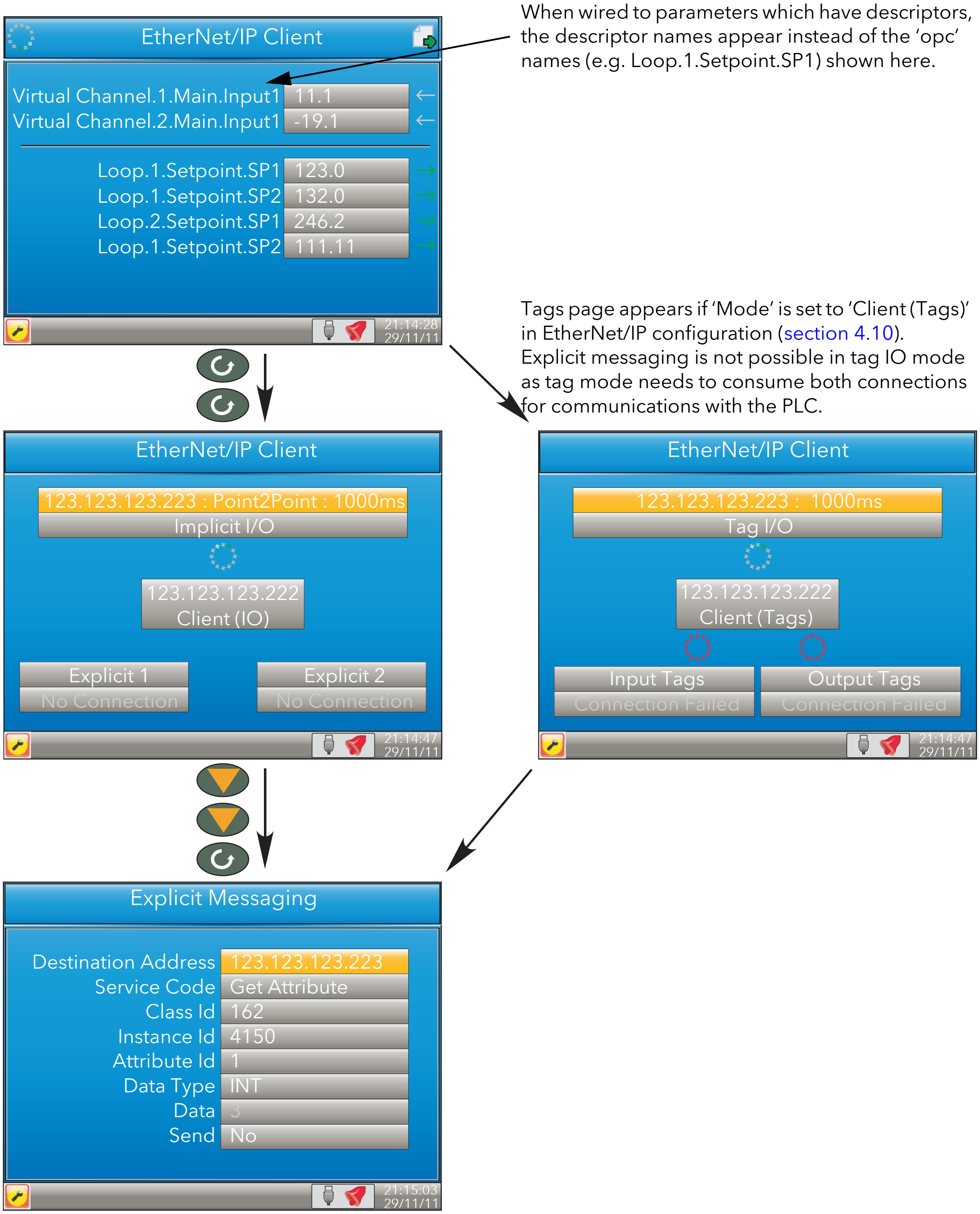



Figure 3.4.13e explicit messaging example

### 3.4.13 ETHERNET/IP DISPLAY MODE (Cont.)

#### USING TAGS

When acting as servers, many PLCs present their data in a tag format instead of implicit data format. For this reason, when the client is configured as 'Client (Tags)', (section 4.10) 30 input and 30 output tags become available to the user via iTools (figure 3.4.13f).

This allows tag names to be typed in, input tags 1 to 30 being associated with implicit inputs 1to 30 respectively and output tags 1 to 30 being associated with implicit outputs 1to 30 respectively.

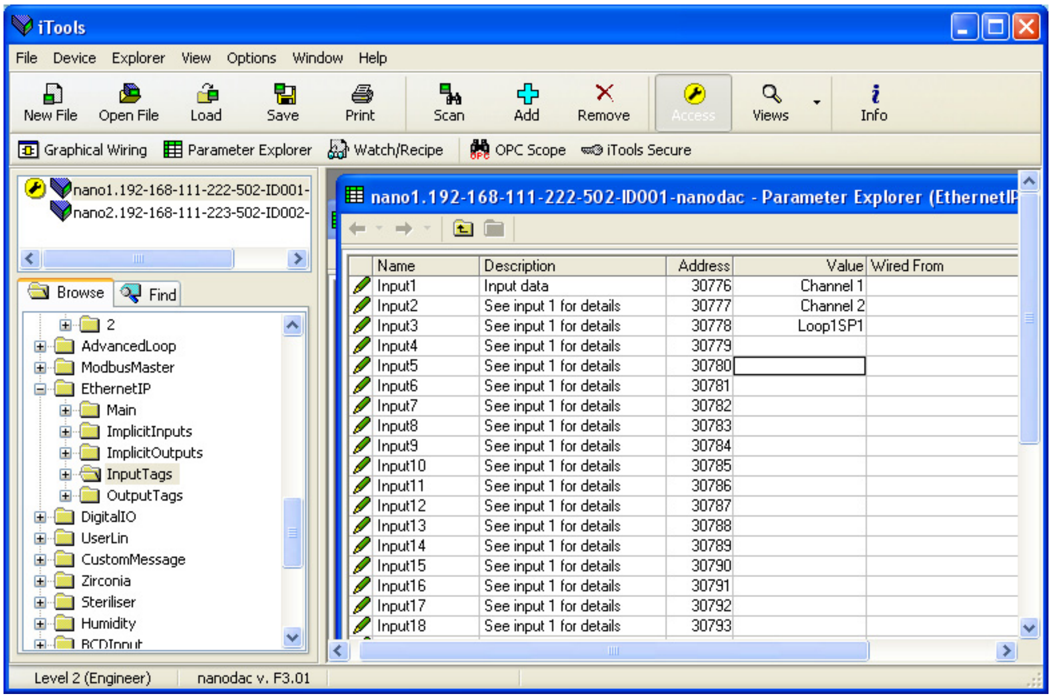



Figure 3.4.13f iTools display showing input tags.

In the example above, the value of the parameter with the tag 'Channel 1' will be written to implicit input 1.

Notes:

1. Most PLCs have a data buffer limit of 500 Bytes. The total number of bytes being used is given by the equation: Total number of data bytes = (tag length + 10) × the number of requested tags.
2. Input data direction is always to the nanodac:
   in server mode input data is written to the nanodac from the client
   in client mode, input data is read by the nanodac from the server device.
3. Output data direction is always from the nanodac:
   in server mode output data is written to the client from the nanodac
   in client mode, output data is read by the server from the nanodac.

## 3.5 TREND HISTORY

Entered from the top level menu (section 3.1), this allows vertical and horizontal traces to be reviewed for Trend group channels. The amount of data displayed in one screen depends on the 'Zoom In/Out' setting in the History menu (section 3.5.2) and on the recording interval selected in Group Recording configuration (section 4.3.2). It is also possible to enter a time and date to which the history then jumps.

The history display is identical in appearance with the trend display except:

1. History displays can include messages if so configured in the History menu.
2. For horizontal trends, the scale is displayed permanently at the left edge of the display.

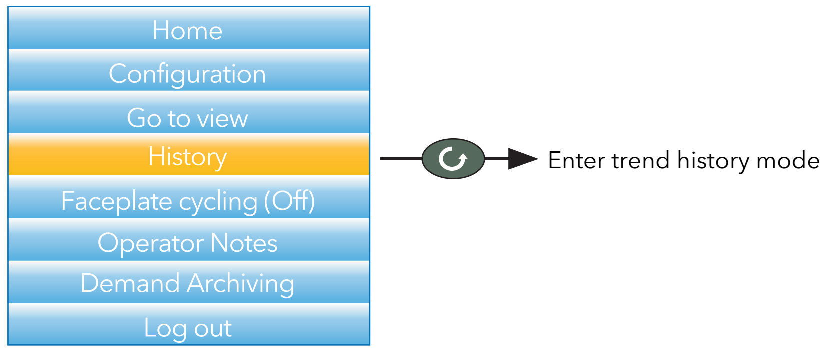



Figure 3.5a Top level menu

### 3.5.1 Navigation

 The down arrow button moves the display backwards in time by $^1/_3$ screen-full per operation (assuming that the current display is not the earliest). See also 'SEARCH FOR', below.

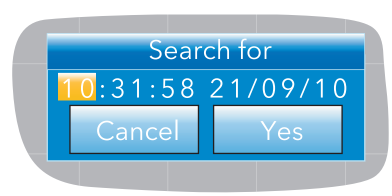
 The up arrow button moves the display forwards in time by $^1/_3$ screen-full per operation (assuming that the current display is not the latest). See also 'SEARCH FOR', below

 The scroll key scrolls through the trend group channels, emphasizing each channel (and displaying its faceplate) as it is selected.

 The page key calls the History Menu, described in section 3.5.2, below.

**SEARCH FOR**

In the history display, holding the up or down arrow key operated for approximately two seconds produces a 'Search for' display which allows the user to enter a time and date. Once a time and date have been entered, 'Yes' then causes the history display to jump to that time and date (if such history exists).




To enter a time and date:

1. Use the up/down arrows to highlight the item to be edited.
2. When highlighted (orange background), operate the scroll button. The highlighted text turns black.
3. Use the up and down arrow keys to scroll to the required value for the field, then operate the scroll button again. The text goes white.
4. Repeat the above editing process for all the remaining items which are to be edited.
5. Use the up/down keys to select 'Yes'. The 'Search for' window closes, and the history display jumps to the selected time and date.

Notes:
1. If no history exists for the selected time and/or date 'No History Available' is displayed.
2. The time and date format and Daylight Savings Time (DST) effects are as set in the 'Locale' area of Instrument configuration. See section 4.1.2 for further details.

### 3.5.2 History Options Menu

Operating the page key from within a history display, causes the History Options menu to appear.

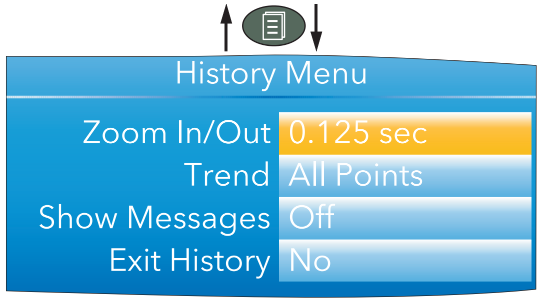



Figure 3.5.2 History Options menu

**PARAMETERS**

| | |
|---|---|
| Zoom In/out | Allows the user to select the amount of history displayed on the screen. |
| Trend | Select either ‘All Points’ or ‘Each Point’.<br>‘All points’ displays all channels in the trend group, with the first channel emphasized on the screen and its faceplate displayed. The Scroll button is used to select the next channel in the group.<br>‘Each Point’ initially displays only the first point in the trace group. The scroll key is used to cycle through individual group channels in turn. |
| Show Messages | ‘Off’ disable the inclusion of messages in history display. ‘On’ causes messages to appear, superimposed upon the point traces (vertical trend mode only). |
| Exit History | Selecting ‘Yes’ for this item causes a return to the top level menu or to the message summary page. |

Note: Operating the page key from the History menu causes a return to the history display.

## 3.6 TEXT ENTRY

The user is often required to enter text characters or numbers (when editing operator notes, for example). This is done using the pop-up keyboards which are displayed when required. When only numerals are required a special keyboard is presented which contains only numerals.

Figure 3.6 shows the three standard keyboards, along with a 'scan' direction for operations of both up arrow and down arrow keys. To change keyboards, use the arrow pushbuttons to highlight the keyboard name ('Numeric', 'Symbols' or 'Alpha'), and then operate the scroll button.

Generally, to enter text, the required character is highlighted using the up and down arrows and the scroll button is used as an 'Enter' key. Once text entry is complete, the Page button is used to confirm the edit (use the down arrow to select 'Yes' then operate the scroll button).

Pressing and holding the scroll button and then immediately operating the up or down arrow, causes the character insertion point to move to the left (down arrow) or to the right (up arrow).

The user can press and hold the scroll key to display variations on certain characters (the letter 'e' in the figure). Once displayed, the up and down arrows can again be used to scroll through auxiliary list, allowing capital letters, and characters with diacriticals (e.g. accents, umlauts, tildes, cedillas) to be selected and entered using the scroll button.

The backarrow key is used as a back space key - i.e. it deletes the character to the left of the cursor position. The 'Del' key deletes the character to the right of the cursor.

Note: Leading and trailing space characters are automatically removed from text strings.

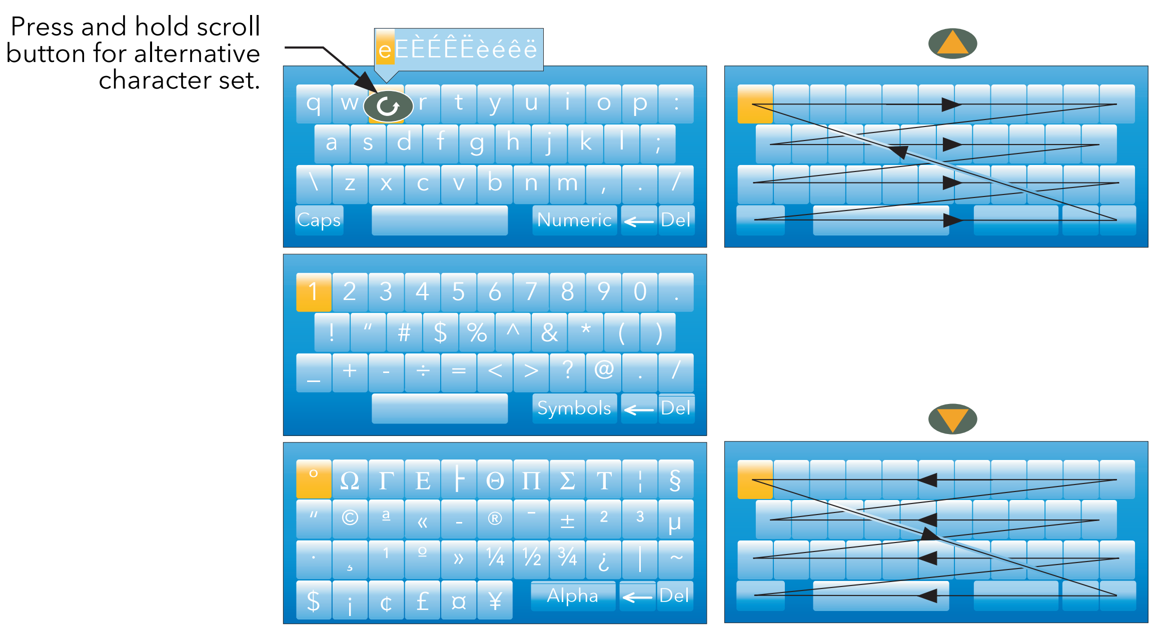



Figure 3.6 Standard Keyboards

### 3.6.1 Numeric keyboard

As mentioned previously, for functions which can take only numerals, a special numeric keyboard appears, as depicted in figure 3.6.1.


Figure 3.6.1 Numeric keyboard

### 3.6.2 USB keyboard

Text and numeric entry can also be carried out using a USB keyboard as described in section 8.3.

# 4 CONFIGURATION

Entered from the top level menu (section 3.1) this allows the recorder configuration to be accessed and edited ('Engineer' access level required for full editing).

**CAUTION**

Recording is stopped for as long as the recorder login is at Engineer level. This means that Input/output circuits are switched off during configuration.

As shown in figure 4, below, the recorder configuration is arranged in a number of 'areas', each of which is allocated its own sub-section within section 4.

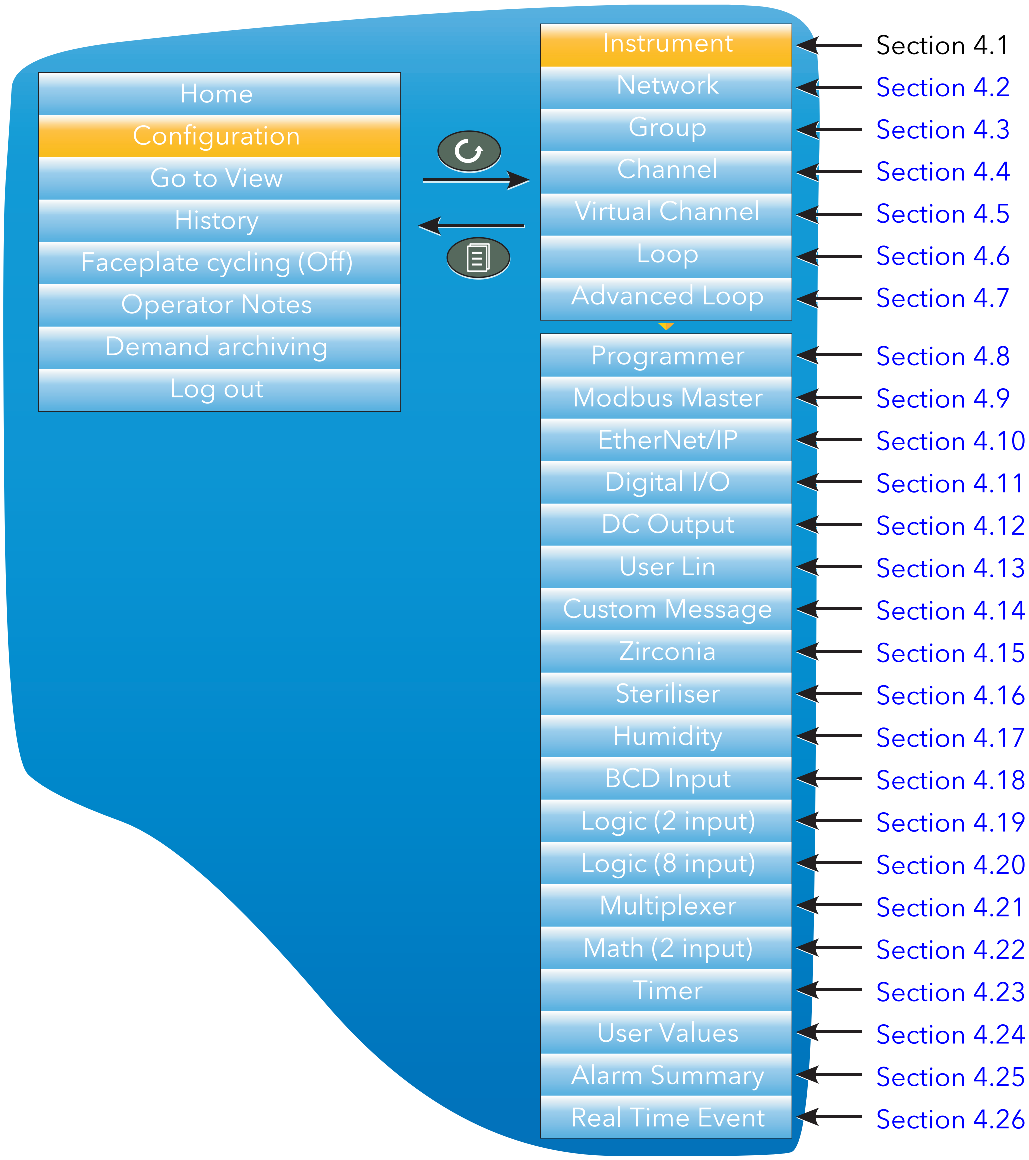



Figure 4 Top level configuration menu

The factory default configuration can be returned-to, if required, by entering a special Engineer password, as described in section 4.1.6.

## 4.1 INSTRUMENT MENU

Instrument

- Clock (section 4.1.1)
  - Date
  - Time
- Locale (section 4.1.2)
  - Language
  - Date format
  - Time zone
  - DST enable
- Display (section 4.1.3)
  - Brightness
  - Save after
  - Saver Brightness
  - Home page
  - HPage timeout
  - Vertical trend
  - Horizontal trend
  - Future trend
  - Vertical bar
  - Horizontal bar
  - Numeric
  - Alarm panel
  - Loop control
  - Dual loop
  - Cascade
  - Programmer
  - Steriliser
  - Promote list
  - Modbus Master
  - EtherNet/IP
  - Trend back-ground
  - History back-ground
  - H.Trend scal-ing
  - Faceplate cy-cling
  - Setpoint colour
  - Number format
  - USB auto scan
- Info (section 4.1.4)
  - Name
  - Type
  - Version
  - Bootrom
  - Company ID
  - Config Rev
  - Security Rev
  - Nvol writes
  - Line Voltage
  - Wires Free
- Upgrade (section 4.1.5)
  - Type
  - IP Address
  - Username
  - Password
  - Source Path
  - Initiate
- Security (section 4.1.6)
  - Engineer Password
  - Supervisor Password
  - Operator Password
  - Feature Pass-words
  - Comms Pass-word
  - OEM Pass
  - OEM Entry
  - OEM Status
- I/O Fitted (Section 4.1.7)
- Save/Restore (Section 4.1.8)
  - Select Save or Restore
- Input Adjust (section 4.1.9)
  - Select channs
  - Start IP adjust
  - Abort
- Output Adjust (section 4.1.10)
  - Output
  - Measured o/p
  - Confirm

### 4.1.1 Clock

The up and down arrows are used to highlight 'Date' (default) or 'Time'.

To set the date, the scroll button is used to display the numeric keyboard described in section 3.6.1. The up and down arrows are used to highlight the relevant numeral or separator ('/' or ':') and the scroll key used to enter it into the display window.

To set the time, the scroll button is operated to enter edit mode, then the up and down buttons are used to scroll to display a time, say 15 seconds later than the current time. Once the current time matches the display, the scroll button is pressed to confirm the time and to start the clock.

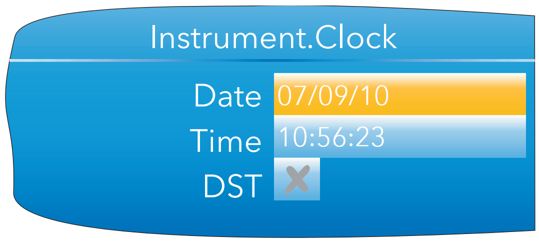



Figure 4.1.1 Clock menu

The 'DST' field appears only If 'DST Enable' is selected 'Yes', in 'Locale' (section 4.1.2). If the 'box' contains a cross (as shown) then Daylight Saving Time (DST) is not currently active. A 'tick' means that the time shown has been advanced by an hour because DST is active.

### 4.1.2 Locale

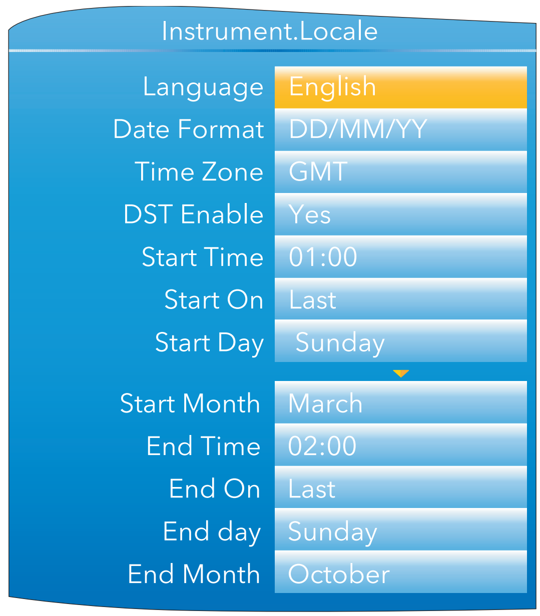



Figure 4.1.2 Typical Instrument configuration menu (expanded to show all fields)

| | |
|---|---|
| Language | Select the language to be used for displays etc. |
| Date format | Select MM/DD/YY, YY/MM/DD as the required format. |
| Time Zone | Select the required offset from GMT (UTC). This setting affects only the displayed time. Archiving, recording etc. times remain in GMT. |
| DST Enable | Daylight Saving Time enable. Once the selection is enabled, the following (previously hidden) fields appear, allowing the start and end dates for Daylight Saving Time (DST) to be configured. DST affects only the displayed time. Archiving, recording etc. times remain in GMT. |
| Start Time | Appears only when 'DST Enable' (above) is set to 'Yes'. Use the up/down keys to scroll to the required start time. |
| Start On | Select 'Last', 'First', 'Second', 'Third' or 'Fourth' as the required week. Used in conjunction with the 'Start Day' and 'Start Month' entries following. |
| Start Day | Select the day of the week on which DST is to commence. |
| Start Month | Select the month in which DST is to commence. |
| End Time, End On, End Day, End Month | As for 'Start Time' etc. above, but specifies the end time and date for daylight savings. |

### 4.1.3 Display configuration

This allows the user to set display brightnesses and screen saver details, to select a display mode as the 'Home' page, and to enable/ disable the various display modes. The normal 'Select, Scroll, Enter' editing technique is used as has been previously described.

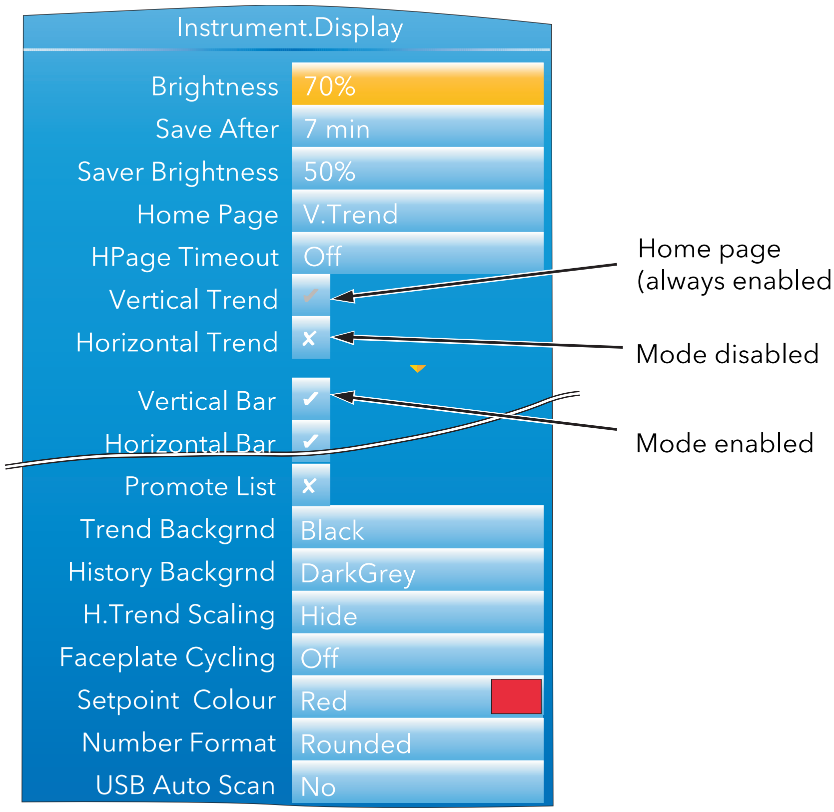



Figure 4.1.3 Display menu (expanded to show all fields)

| | |
|---|---|
| Brightness | Allows the user to select a normal operating brightness for the screen from 10% to 100%, in 10% steps. |
| Save After | The elapsed time (since last button press) before the screen switches from 'Brightness' to 'Saver Brightness'. (Off = saver function disabled) |
| Saver Brightness | The screen saver brightness. Valid entries are 10% to 100% inclusive, in 10% steps. Using a lower power when not 'in use' not only saves power, but also increases display life. Typical screen power consumption is 0.5W at 100%, falling in a linear fashion to 0.05W at 10%. |
| Home page | Allows any display mode to be chosen as the 'Home' page. This is the page that the recorder displays at power up, and also the page displayed when the 'Home' key is selected from the top level menu (section 3.3). The selected display mode (vertical trend in figure 4.1.3) is always enabled in the following display mode enable fields (its 'tick' is greyed out and cannot be edited). See section 3.4 for a description of the available modes. |
| HPage Timeout | The elapsed time (since last button press) before the display returns to the home screen. (Off = disabled) |
| Vertical Trend | This is the default home page, and its tick is greyed. If this is not the home page, the tick can be changed to a cross, by highlighting it and operating the scroll button. |

### 4.1.3 DISPLAY CONFIGURATION (Cont.)

Horizontal Trend, Vertical Bar, Horizontal bar, Numeric, Alarm Panel, Loop control, Dual Loop, Cascade, Programmer, Steriliser, Promote List, Modbus Master, EtherNet/IP. As for Vertical Trend, above. By default some display modes are disabled (grey cross). In order to enable such display modes the relevant cross is highlighted using the up/down arrow buttons, and the scroll button then used to change the grey cross to a white tick.
The tick associated with the selected home page is always grey.

Note: Some display modes are available only if the relevant option is fitted.

| | |
|---|---|
| Future Trend | This and the associated colour selections appear only if the Programmer option is fitted. See section 3.4.9 for more details. |
| Trend Background | Allows the user to select black (default), white dark grey or light gray as the ‘chart’ colour. |
| History Background | As above for ‘Trend background’, but for history displays. |
| H.Trend Scaling | As described in section 3.4.2, by default, the scale for horizontal trends appears at the left edge of the chart for a few seconds before the chart expands leftwards to occupy the scale area. Setting ‘H.Trend Scaling’ to ‘Permanent’, ensures that the scale remains permanently on display. |
| Faceplate cycling | Allows the default faceplate cycling state to be defined as ‘On’ or ‘Off’ (section 3.3.5) |
| Setpoint colour | The colour for the setpoint in Control Loop display pages (section 3.4.7). |
| Number Format | Rounded:<br>Truncated: |
| USB Auto Scan | If set to ‘Yes’, bar code data messages are automatically generated and appear on the display and in the Message list without operator intervention. If set to ‘No’, the Message appears on the screen for editing and/or confirmation, before being displayed etc. Section 8.2 provides further details. |

There is a new paramter been added to the Instrument.Display list - Number format.

The options are to "Round" or "Truncate" values. On the previous phases of the nano, numbers were truncated (in the same was as the 6000).

With phase 3 there is an option to allow numbers to be rounded. The reason for this is driven primarily from a control point-of-view. With truncation, it is quite likely that the PV will look as though it never settles onto setpoint. The rounding/truncation affects the UI display and MODBUS scaled integers, the underlying numbers are not affected, nor the values saved in the history files. Over MODBUS comms, all floating point parameters that are read via scaled integer comms will take note of the configured setting for rounding or truncating and reflect this. On the UI, ALL floating point values rendered will adhere to the configured setting of rounding or truncating.

### 4.1.4 Info menu

Gives information about the recorder hardware and software, and allows the user to enter a descriptor for the instrument. The normal 'Select, Scroll, Enter' editing technique, previously described) is used to edit those fields that are not read only.

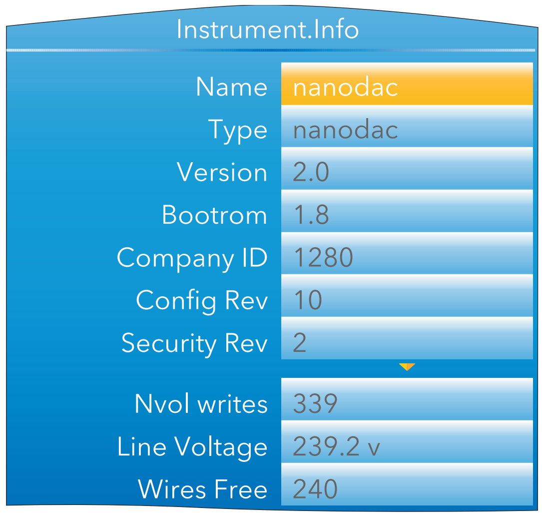



Figure 4.1.4 Info menu (expanded to show all fields)

| | |
|---|---|
| Name | Allows the user to enter a descriptor of up to 20 characters, using the text entry techniques described in section 3.6. The number of characters visible in the display mode pages varies according to the number of alarm symbols on display. |
| Type | Nano. Read only display of the instrument model (used by 'iTools'). |
| Version | Read only. The software version of the instrument. |
| Bootrom | Read only. Instrument software Boot ROM version |
| Company ID | Read only. For CNOMO* purposes over Modbus (1280 decimal; 0500 hex). |
| Config Rev | Read only. This value is updated, and a message including this value generated, every time configuration is quit, if any one or more configuration parameter has been changed. |
| Security Rev | Read only. This number is incremented every time configuration is quit, if any one or more passwords has been changed, or if the FTP Server username has been changed, or if the Comms Enable field has been edited. |
| Nvol writes | Number of non volatile write operations for diagnostic purposes. |
| Line voltage | The instantaneous value of the supply voltage applied to the instrument. Used in some control loop operations. |
| Wires Free | This shows the number of wires free to be used. The value takes into account all user wiring whether carried out at the instrument or downloaded from the iTools graphical wiring editor. |

* CNOMO = Comité de normalisation des moyens de production.

### 4.1.5 Upgrade

**CAUTION**

1. Power must not be removed from the unit whilst upgrade is in progress, as to do so will cause permanent damage to the unit.
2. For USB upgrades, the memory stick must not be removed whilst upgrade is in progress or the instrument will be permanently damaged.

This item allows the user to update the instrument firmware, either from a memory stick in the USB socket at the rear of the unit, or via FTP transfer from a host computer. Firmware upgrade files are downloaded from the recorder manufacturer and transferred to the instrument by memory stick or by FTP transfer. Splash screens are prepared by the user and transferred using a memory stick. The unit restarts automatically after an upgrade or splash screen replacement.

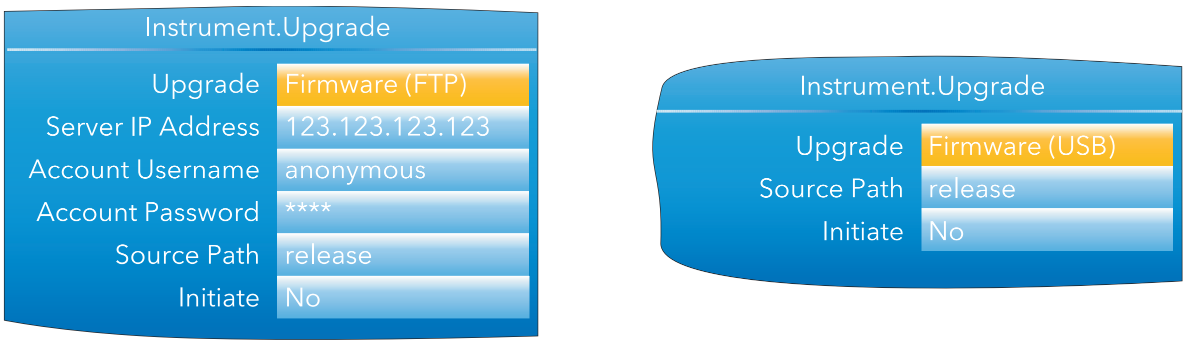



Figure 4.1.5 Typical Upgrade menus

| | |
|---|---|
| Upgrade | Select 'Firmware (USB)', 'Firmware (FTP)', 'Bootrom (USB)' or 'Splash (USB)' as the source of the upgrade. |
| Server IP Address | For 'Upgrade' = 'Firmware (FTP)' only, this field must contain the IP address of the pc which is to supply the upgrade file. |
| Account Username | For 'Type' = 'Firmware (FTP)' only, the username set up in the host ftp server |
| Account Password | For 'Type' = 'Firmware (FTP)' only, the password set up in the host ftp server |
| Source Path | The name of the directory from which the upgrade file is to be read. This is only the name of the directory without any path elements (e.g. '/') included unless the path is 'release/upgrade/files'. |
| Initiate | Select 'Yes' to initiate the upgrade. |

#### CUSTOMISING THE SPLASH SCREEN

'Splash (USB)' allows the user to select a new image for the splash screen (i.e. the screen that appears at power up or restart). When 'Initiate' is set to 'Yes', the instrument searches the USB device for a file called 'splash.bmp' located in the 'release' folder. If such a file is found, it is loaded, and the instrument re-starts with the new image as the 'splash' screen. If no file is found, the request is ignored. If the image is not of the correct type or size, the instrument re-starts with the default splash screen.

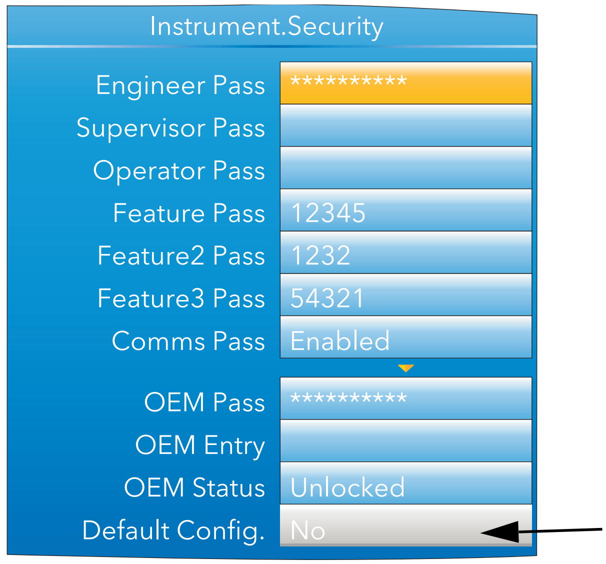
The original splash screen is included on the 'tools' DVD, so that it can be restored if required.

Rules:

1. This feature is available only with Bootrom versions 2.0 and above.
2. The file must be located in a folder called 'release' and the file name must be 'splash.bmp'.
3. The image must be 320 x 240; 24-bit resolution.
4. The image must be in bitmap (suffix.bmp) format.
5. The image may not exceed 256kB.

### 4.1.6 Security menu

This allows the user to enter passwords for all security levels (except logged out), and to enable/disable serial communications security.




Appears only if Engineer Password = reset

Figure 4.1.6 Security menu

| | |
|---|---|
| Engineer Pass | Gives access to configuration menus. Set to 100 when despatched, but can be edited here, if required, by entering an alternative of up to 20 characters (note 1).<br>If 'reset' (case sensitive) is entered as the Engineer Password, the 'Default Config.' field appears allowing the instrument default configuration to be restored (note 2). |
| Supervisor Pass | A password (none by default) of up to 20 characters can be entered here to protect Supervisor level access. |
| Operator Pass | A password (none by default) of up to 20 characters can be entered here to protect Operator level access. |
| Feature Pass | This is a password supplied by the manufacturer to enable the software options (e.g. Loop, Zirconia block, Toolkit blocks etc.). When applying for this code, the manufacturer will require the instrument's MAC address (Network.Interface menu section 4.2.1) and the instrument's firmware Version (Instrument.info menu - section 4.1.4). The password is MAC address dependent so that it cannot be used on any other instrument. |
| Feature2/3 Pass | Similar to 'Feature Pass' above, but for additional features. |
| Comms Pass | Enables/disables password security for external communications (including via iTools). If set to 'Enabled', the Engineer level password will be required if an attempt is made to enter the configuration menus from a remote pc. If set to 'Disabled', then access to configuration can be gained over a communications link, without a password.<br>If enabled, then entry to configuration mode via the Instrument Mode (IM) parameter must be completed within 5 seconds of entering the password, or the attempt will fail. |

Notes:

1. It is recommended that only such characters as appear on the user's pc keyboard be used in the Engineer password. The use of other characters makes it necessary to use 'Escape' codes (e.g. Alt 0247 for the '÷' sign) when trying to enter configuration mode from iTools, for example.
2. Restoring factory default configuration can also be carried out in iTools, using the Engineer password 'reset' and selecting Default Config to 'Yes'.

### 4.1.6 SECURITY MENU (Cont.)

| | |
|---|---|
| OEM Pass | The configured pass phrase used to enable / disable the OEM security option. This field is editable whilst the OEM Status is 'Unlocked' and the user has 'Engineer' access. |
| OEM entry | To lock or unlock the OEM security feature, the user must enter the pass phrase entered in 'OEM Pass' above. |
| OEM Status | Read only 'Locked' or 'Unlocked' status display. |
| Default Config | This field appears only if 'reset' has been entered as the Engineer Password. Selecting 'Yes' Causes the instrument to restart with default configuration (i.e. the instrument 'cold starts'). See note 2 above. |

#### OEM SECURITY

In products that incorporate user wiring, the value of an application may lie more in the user wiring (connecting the function blocks together) than in the configuration of the instrument's parameters.

OEM Security allows the user to prevent the application from being copied either via comms (by iTools or a third party comms package) or via the instrument's user interface.

When OEM security is enabled, users are prevented from accessing wiring (for reading or writing) from any source (comms or user interface), and it is not possible to Load or Save the configuration of the instrument via iTools or by using the Save/Restore facility (section 4.1.8).

### 4.1.7 I/O fitted

This provides a read only display showing what type of input or output circuit is associated with each set of rear terminals.

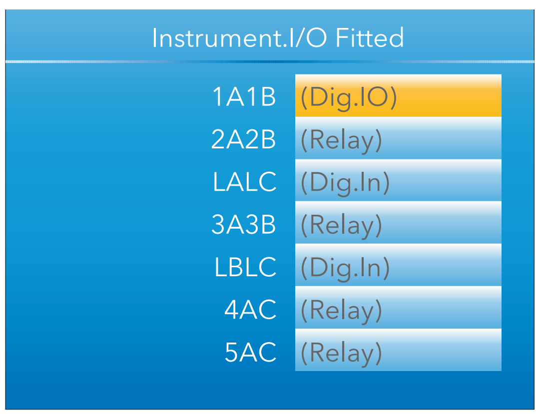



Figure 4.1.7 I/O fitted display

#### I/O TYPES

| | |
|---|---|
| Dig.IO | Digital input/output |
| Relay | Relay output |
| Dig.In | Digital input |
| Dig.Out | Digital output |
| DC.Op | DC output |

Note: The I/O types fitted in locations LALC, LBLC, 4AC and 5AC are always as shown above. The types of I/O fitted in locations 1A1B, 2A2B and 3A3B depends on the options specified at time of order.

### 4.1.8 Save/Restore

This allows the user to save and/or restore instrument clone configurations to a memory stick inserted into the USB connector at the rear of the unit. The format of the saved/restored files is iTools clone files (*.uic)

Selecting 'Restore' presents a list of clone files in the configured directory on the USB device. (In the example below, the file is located in the basic usb0 directory - it has not been saved to a particular configuration directory.)
When 'Save' is selected, the virtual keyboard must be used to enter the filename. If the file already exists on the USB device, a warning appears offering 'Cancel' or 'Overwrite' alternatives.

Notes:
1. The ability to save and restore is disabled if OEM security is enabled.
2. Configuration save/restore is available only when the unit is logged into at 'Engineer' access level.
3. During USB cloning (USB save/restore), the priority of modbus slave comms is lowered. This allows the save/restore process to complete in a minimal time (around 60 seconds). During this period, modbus slave comms response times will be extended and may result in the master device timing-out.

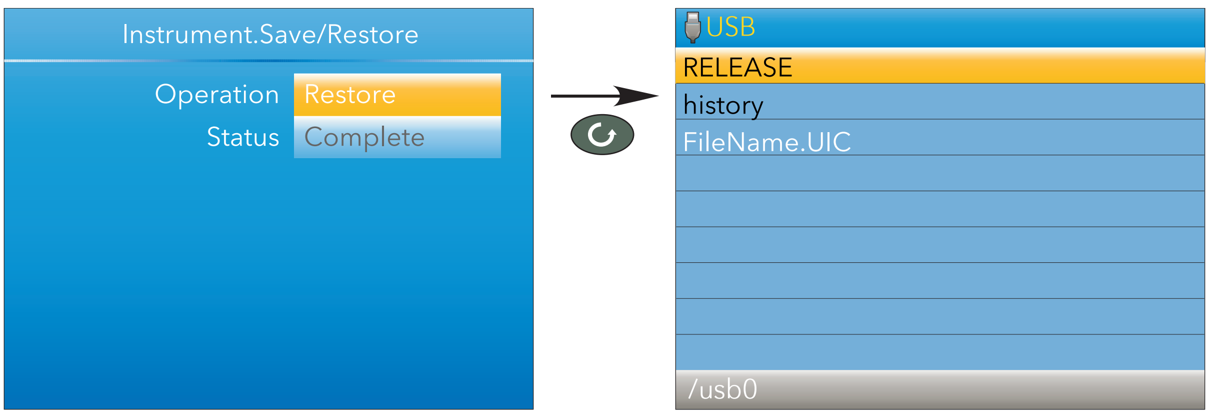
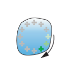



Figure 4.1.8 Save/Restore display

| | |
|---|---|
| Operation | Select 'Save' or 'Restore'. Use the up/down arrow keys to highlight the required .UIC file, then use the scroll key to initiate the operation. |
| Status | Shows the status of the operation, as follows:<br>Inactive: Neither saving or restoring a clone file has occurred since the last time the instrument was power cycled.<br>Complete: Indicates that the cloning process has completed.<br>Restoring: Restore operation is currently in progress.<br>Saving: A clone file is currently being saved.<br>Cold started: A power-cycle of the product occurred whilst a Restore operation was in progress. The product configuration is unreliable and has been reset to factory default. |



The 'Restoring' and 'Saving' status text is accompanied by an animated display (circling green 'flash') to indicate that the operation is in progress.

### 4.1.9 Input adjust

Notes
1. Input adjust cannot be applied to input channels with input type of 'Digital', 'Test' or 'Off'.
2. Input adjustments can be carried out only by users logged in as 'Engineer' (see section 3.3.7).
3. The instrument must be powered for a sufficient time (e.g. 30 minutes) for it to reach thermal equilibrium before an input adjust is performed.

This facility allows the user to compensate for tolerance errors etc. The technique used is to select those channels to which adjust is to be applied, then for each channel to:

a apply a known low level signal (at or close to the low input range value) to the relevant input. When the recorder reading is steady, press 'Apply'.

b. apply a known high level signal (at, or close to, the high input range value) to the relevant input. When the recorder reading is steady, press 'Apply'.

Figure 4.1.9a shows a typical display when 'Input adjust' is selected from the Instrument menu, and Apply adjust has been selected. As can be seen, channel 3 has previously been adjusted.

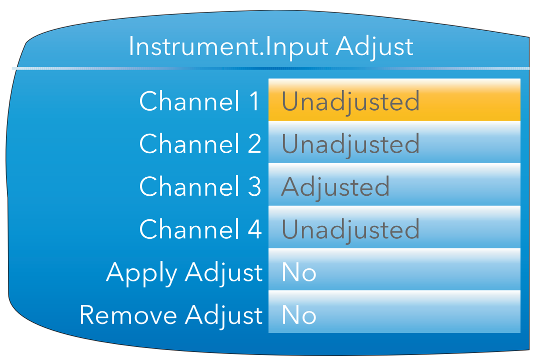



Figure 4.1.9a Input adjust top level display

| | |
|---|---|
| Channel 1 to 4 | Shows the adjust status of each channel |
| Apply Adjust | Selecting 'Yes' initiates the adjustment procedure described below. |
| Remove Adjust | Selecting 'Yes' initiates the adjustment removal procedure described below. |
| Abort | Allows the user to abandon input adjustment at any point in the procedure. |

**ADJUSTMENT PROCEDURE**

1. As shown in figure 4.1.9b, highlight the 'Apply Adjust' field, and operate the scroll key to enter edit mode. Use the up or down arrow key to select 'Yes'. Use the scroll button to change Channel 1 'cross' to a 'tick' (check mark). Similarly select any other channels which require adjustment.

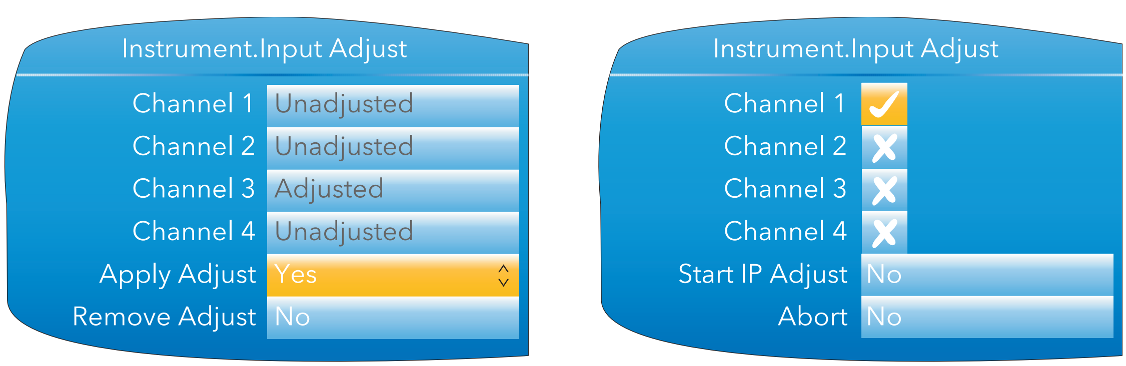



Figure 4.1.9b Channel adjustment procedure (1)

### 4.1.9 INPUT ADJUST (Cont.)

**ADJUSTMENT PROCEDURE (Cont.)**

2. Highlight the 'Start IP 'Adjust' field and use the scroll and up/down arrow to select 'Yes'. Use the scroll key again to enter the low value adjust page.
3. Apply the known low value and wait for the value to stabilise. Enter the 'Low Target Value' (the value that the recorder is to read for the applied input). When all is steady, use the scroll and up/down arrow to set the 'Confirm Low' field to 'Yes', then operate the scroll button again.

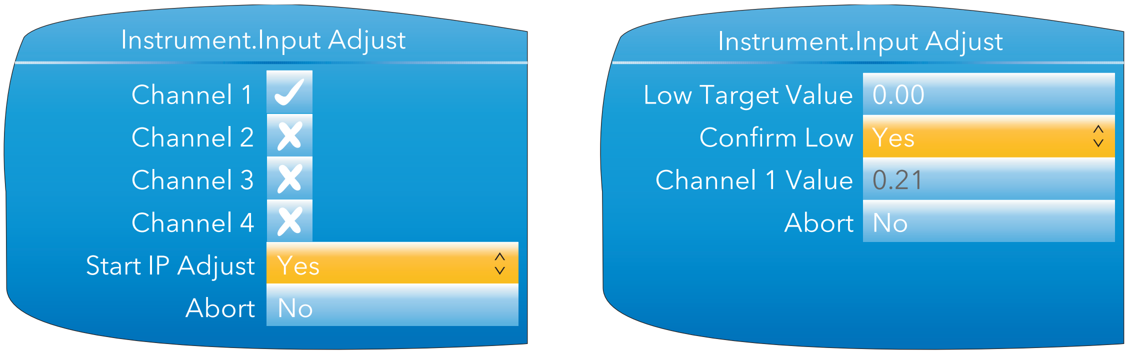



Figure 4.1.9c Channel adjustment procedure (2)

4. The display changes to the high value adjust page.
5. Apply the known high value and wait for the value to stabilise. Enter the High Target Value (the value that the recorder is to read for the applied input). When all is steady, set 'Confirm High' to 'Yes'.

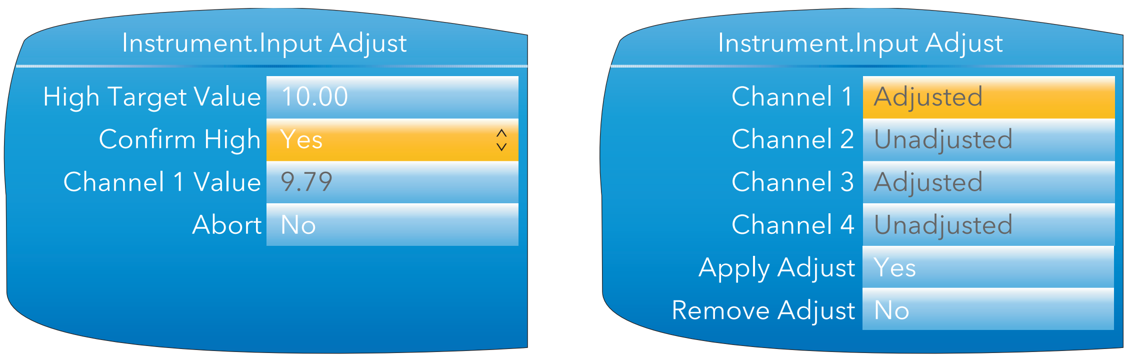



Figure 4.1.9d Channel adjustment procedure (3)

**REMOVAL PROCEDURE**

1. Set 'Remove Adjust' to 'Yes' and operate the scroll button.
2. Use the scroll and up/down arrow buttons to change the required channel icons from crosses to ticks.
3. Select Remove IP Adjust to 'Yes' and operate the scroll key. The adjustment is removed from all selected channels without further confirmation.

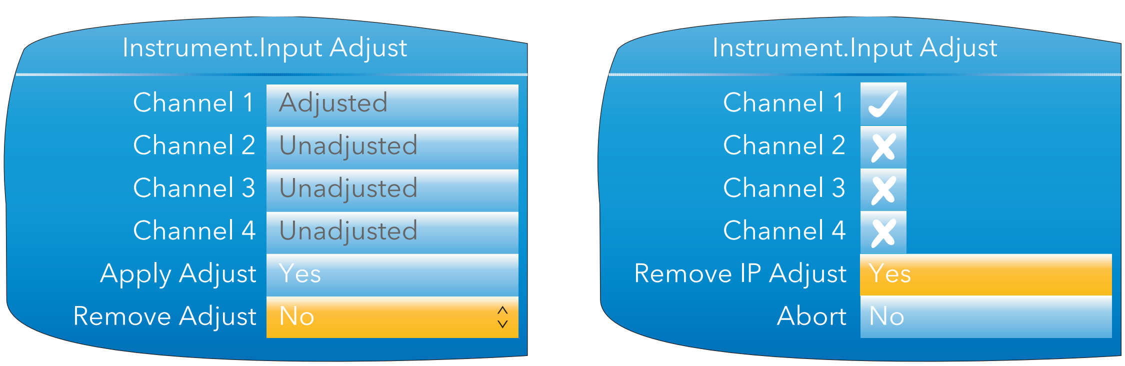
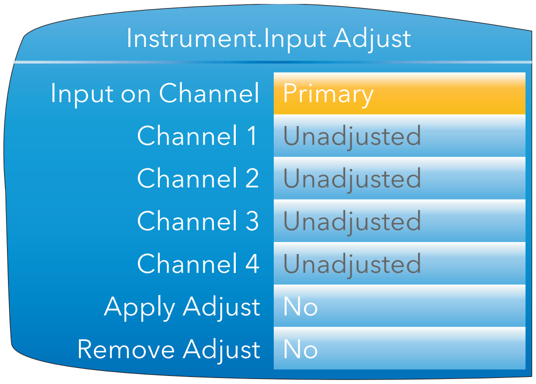


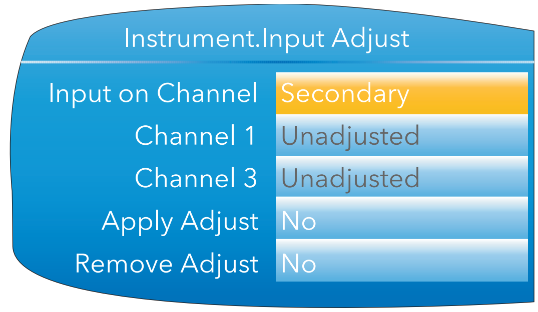

Figure 4.1.9e Channel adjustment removal

### 4.1.9 INPUT ADJUST (Cont.)

#### DUAL INPUT CHANNELS

For the dual input channel option, input adjust is carried out as described above, except that for any channel where dual inputs are configured, the user must initiate adjustment to primary and secondary inputs separately. As shown in figure 4.1.9f, a new field 'Input on Channel' is introduced for this purpose.



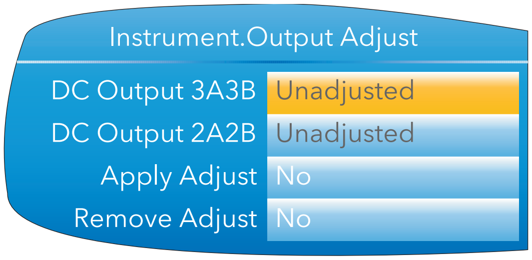

Only those channels with 'Type' set to 'Dual mA', 'Dual mV' or 'Dual T/C' appear in the list of secondary channels. In this example, only channels 1 and 3 are configured as dual input. (See section 4.4.1 for channel Type configuration.)

Figure 4.1.9f Input adjust top level display (dual input channels)

For primary inputs, all four channels are included in the list and can therefore be selected for adjustment. For secondary inputs, only those channels which have been configured as dual input are included.

### 4.1.10 Output adjust

This item appears only if one or more of I/O type DC Output is fitted and allows the user to compensate for tolerance errors etc. in connected equipment.




1A1B and 2A2B can be configured only as mA outputs.

3A3B can be configured as mA or Volts.

See section 4.12 for configuration details.

Figure 4.1.10a Output adjust initial display

**ADJUST PROCEDURE**

1. Highlight the 'Apply Adjust' field, and operate the scroll key to enter edit mode. Use the up or down arrow key to select the required output and confirm with the scroll key. The output adjust page appears for the low point.
2. Measure the output at the required point, and enter this value in the 'Measured Output' field using the text entry techniques described in section 3.6. To skip this stage go to step 3.
3. Set 'Confirm Low' to 'Yes'. The output adjust page appears for the high point.
4. Measure the output at the required point, and enter this value in the 'Measured Output' field as described for the low point. To skip this stage go to step 5.
5. Set 'Confirm High' to 'Yes'. The output adjust initial display reappears, with the word 'Adjusted' in the relevant DC Output field.

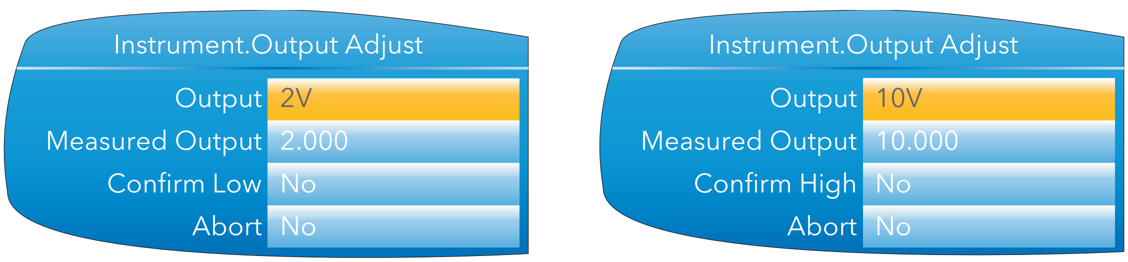



Figure 4.1.10b Low and High adjust point displays

Notes:

1 The figures above show the displays when the DC output is set to 'Volts' (section 4.12) (3A3B only). The mA displays are similar, but the fixed low and high values are 4mA and 20mA respectively.

2. 'Abort' cancels operations so far and returns to the output adjust initial display (figure 4.1.10a).

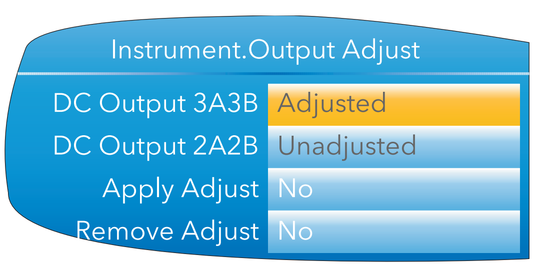



Figure 4.1.10c Adjusted display

**ADJUST REMOVAL**

In the output adjust initial display (figure 4.1.10c) highlight the 'Remove Adjust' field, and operate the scroll key to enter edit mode. Use the up or down arrow key to select the required output and confirm with the scroll key. The output adjustment is removed, without confirmation. The initial display returns to 'Unadjusted' as in figure 4.1.10a.

## 4.2 NETWORK MENU

Network

- Interface (Section 4.2.1)
  - MAC address
  - Client Identifier
  - IP type
  - IP address
  - Subnet mask
  - Gateway
- Archiving (Section 4.2.2)
  - Media size
  - Media free
  - Media duration
  - Rate
  - Destination
  - File format
  - On media full
  - Remote path
  - Primary server
  - Primary user
  - Primary password
  - Secondary server
  - Secondary user
  - Secondary password
  - Trigger
  - Period
- FTP Server (Section 4.2.3)
  - User name
  - Password
- Modbus (Section 4.2.4)
  - Prefmaster IP Address
  - Input timeout
  - Unit ID Enable
  - Serial Mode
  - Time Format
  - Pref Master Conn
  - Response time
  - Master Conn 2
  - Response time
  - Master Conn 3
  - Response time
  - Master Conn 4
  - Response time

### 4.2.1 Interface

This area of configuration allows the user to set up an IP address for the instrument, either by typing one in (Fixed), or automatically (DHCP), assuming a DHCP server is running.

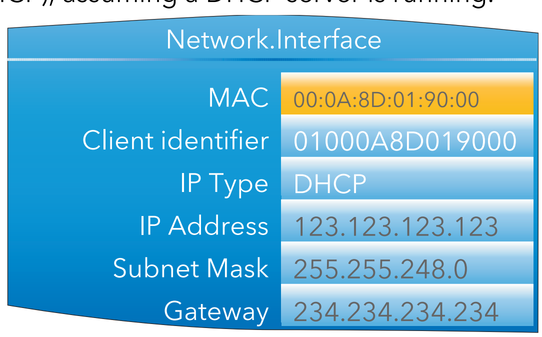



Figure 4.2.1 Network Interface menu

| | |
|---|---|
| MAC | Read only. Media Access Control. A unique address for each instrument, entered at the factory. |
| Client Identifier | The client identifier is a unique id used by DHCP servers that implement option 61. Each nano product will have a unique ID built up from its MAC address. If the DHCP server is configured to use option 61, then it will use this id instead of the MAC address to assign a dynamic IP address. |
| IP Type | If 'Fixed', the user needs to enter an IP address and Subnet Mask in the following fields, and a Gateway address if required.<br>If 'DHCP' the subsequent fields become read only, with the entries automatically generated by the DHCP server. When set to DHCP, it takes several seconds before the IP address is obtained from the DHCP server. |
| IP Address | Read only if 'IP Type' = 'DHCP'.<br>If 'IP Type' = 'Fixed', the user may enter an IP address (IPV4 dot notation). This would normally be supplied by the user's IT department, or from the Network supervisor. |
| Subnet Mask | Read only if 'IP Type' = 'DHCP'.<br>If 'IP Type' = 'Fixed', this sets a range of IP addresses that can be accessed. Normally supplied by the user's IT department, or from the Network supervisor. |

### 4.2.1 INTERFACE (Cont.)

Gateway — Read only if 'IP Type' = 'DHCP'.
If 'IP Type' = 'Fixed' this allows the user to enter a gateway address for use when the unit is to communicate outside the local network. Normally supplied by the user's IT department, or from the Network supervisor.

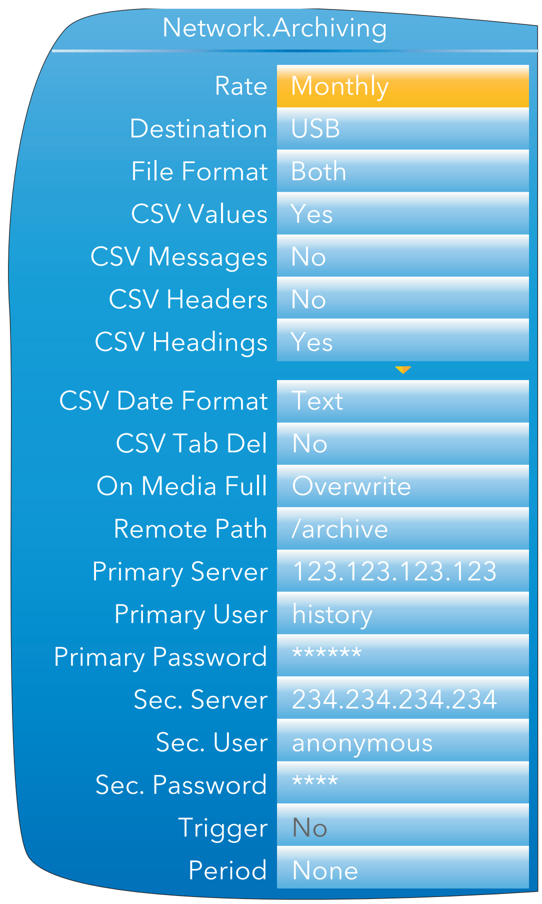
### 4.2.2 Archiving

This area of configuration is used to set up the parameters for use during unattended archiving. Some of the fields appear only if other fields are set to a particular value. For example, the CSV fields appear only if 'File Format' is set to 'CSV' or to 'Both'.

The archived data is not removed from the flash memory of the instrument. When the flash memory is full, new data causes the oldest file(s) to be discarded.

Note: For remote archiving, the host computer must be set up to respond to 'pings'. This is because the nano pings the host whilst establishing connection, and if it does not receive a response the archive attempt fails.


Remote with Binary file format




Local with CSV files included

Figure 4.2.2a Unattended Archive configuration (typical settings)

Media Size — Appears only for File Format = 'Binary (UHH)'. A read only value showing the capacity of the memory stick inserted in the USB port at the rear of the unit. Shows zero if no memory stick is present.

Media Free — Appears only for File Format = 'Binary (UHH)'. A read only value showing the space remaining in the memory stick inserted in the USB port at the rear of the unit. Shows zero if no memory stick is present.

Media Duration — Appears only for File Format = 'Binary (UHH)'. A read only value showing the time it will take to fill the Memory stick if the recorder configuration remains unchanged.

### 4.2.2 ARCHIVING (Cont.)

Rate: Allows the user to specify the frequency at which the contents of the Flash memory are archived to the USB port or, via FTP, to a pc. Scrollable settings are:

- None: Automatic archiving is disabled. Any archiving must be initiated by the user using Demand Archiving, as described in section 3.3.7.
- Hourly: Archive occurs on the hour, every hour.
- Daily: Archive initiated at 00:00* each day
- Weekly: Archive is initiated at midnight* every Sunday
- Monthly: Archive is initiated at 00:00* on the 1st of every month.
- Automatic: The recorder selects the least frequent of the above archive periods which is guaranteed not to lose data as a result of the internal flash memory's running out of space.

*Note: Archive times are not adjusted for daylight saving time (DST). Thus, if the archive is set to 'Daily', 'Weekly' or 'Monthly', then during summer time, the archive will be triggered an hour late (i.e at 01:00 hours instead of midnight).

Destination: Select 'FTP Server' for archive to a remote pc, or 'USB' to archive to the USB port device.

File format: Select 'Binary (UHH)' 'CSV' or 'Both'.

- Binary (UHH): A proprietary format used by the instrument that needs other software (e.g. Review', to interpret the data before it can be presented in spreadsheets etc. Binary files have the extension '.uhh'.
- CSV: This format is a standard open-file format for numeric data. A simple ASCII-based format, it is readable by a wide range of pc applications as well as being suitable for direct import into many commercial databases. CSV files have the extension '.csv'.
- Both: Archiving includes both .uhh and .csv files.

Note: CSV is ASCII based and cannot interpret Unicode characters. For this reason, some characters available to the user will not be displayed correctly in .csv files.

CSV Values: Appears only if 'File Format' is set to 'CSV' or 'Both'. If 'Yes' is selected, then process values are included in the file (see figure 4.2.2b for details).

CSV Messages: Appears only if 'File Format' is set to 'CSV' or 'Both'. If 'Yes' is selected, then messages are included in the file (see figure 4.2.2b for details).

CSV Headers: Appears only if 'File Format' is set to 'CSV' or 'Both'. If 'Yes' is selected, then Header details are included in the file (see figure 4.2.2b for details).

CSV Headings: Appears only if 'File Format' is set to 'CSV' or 'Both'. If 'Yes' is selected, then column headers are included in the file (see figure 4.2.2b for details).

CSV Date Format: Appears only if 'File Format' is set to 'CSV' or 'Both'. Allows 'Text' or 'Spreadsheet' to be selected. Text causes a time/date to appear in the spreadsheet. 'Spreadsheet Nu' displays the number of days since December 30th 1899. The decimal part of the number represents the latest six hours. For example: DDD--- --DD.25 represents 06:00 hours and DDD--- --DD.5 represents 12:00 hours. Spreadsheet Numeric format is more easily interpreted than 'Text' by some spreadsheet applications.

CSV Tab Del: Appears only if 'File Format' is set to 'CSV' or 'Both'.
CSV (Comma Separated Variables) does not always use commas as separators. For example, in some countries the decimal point is represented by a full stop (period), whilst in others a comma is used. In order to avoid confusion between a comma as a decimal point and a comma as a separator, a different separator can be used. This field allows the 'tab' character (^t) to be used instead of a comma.

### 4.2.2 ARCHIVING (Cont.)

On Media Full — For 'Destination' = 'USB' only, this allows the user to select 'Overwrite' or 'Stop' as the action to be taken when the memory stick is full. 'Overwrite' causes the oldest data to be discarded from the memory stick to make room for newer data. 'Stop' inhibits archiving activity.

Remote Path — Left blank if the archive destination is the home folder. If the destination is to a subfolder within the home folder, then the name of the subfolder is entered here, preceded by a '/' character (e.g. '/history').

Primary Server — Allows the user to enter the IP address for the pc to be used as the primary FTP server.

Primary User/Password

These are the Login name and password of the remote host account, assigned either by the Network administrator, or set up in the 'Guest' account of the remote host's 'FTP server' or 'User Manager' configuration.

Sec. Server/user/password

As Primary server details above, but for the secondary FTP server used when the primary is not available for any reason.

Trigger — This parameter can be 'wired' to, say, an alarm going active, or a digital input, to allow an archive to be triggered remotely. Can also be set to 'yes' manually.

Period — Appears only if 'Trigger' is wired (section 7). Allows a period of history to be selected for archiving when 'Trigger' goes 'true. Selections are: None, Last Hour, Last Day, Last Week, Last Month, All, Bring to Date. ('Last Month' archives the last 31 days of history.)

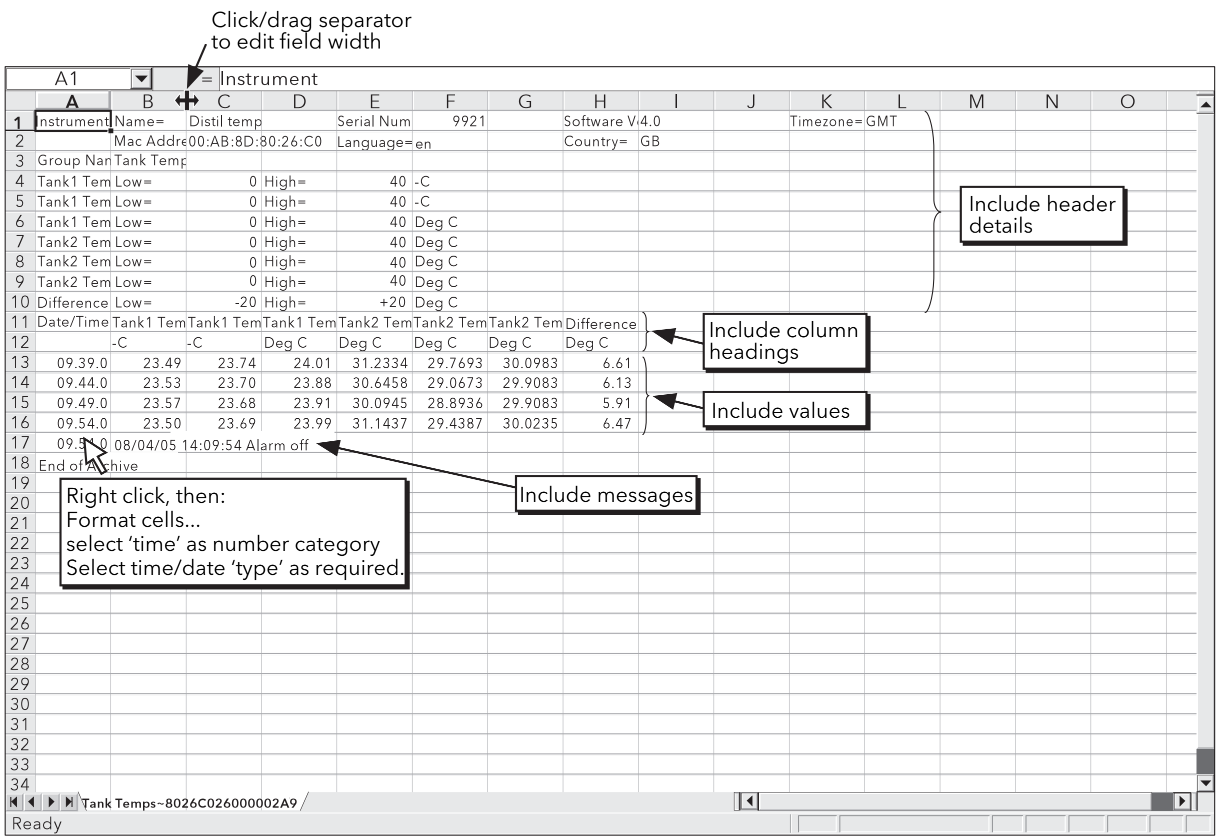



Figure 4.2.2b CSV data example

### 4.2.3 FTP Server

This area of configuration allows the user to enter the Username and Password used to access the instrument from a remote FTP client.

### 4.2.4 Modbus TCP

This allows the user to configure the recorder so as to allow it to communicate using Modbus Transmission Control Protocol.

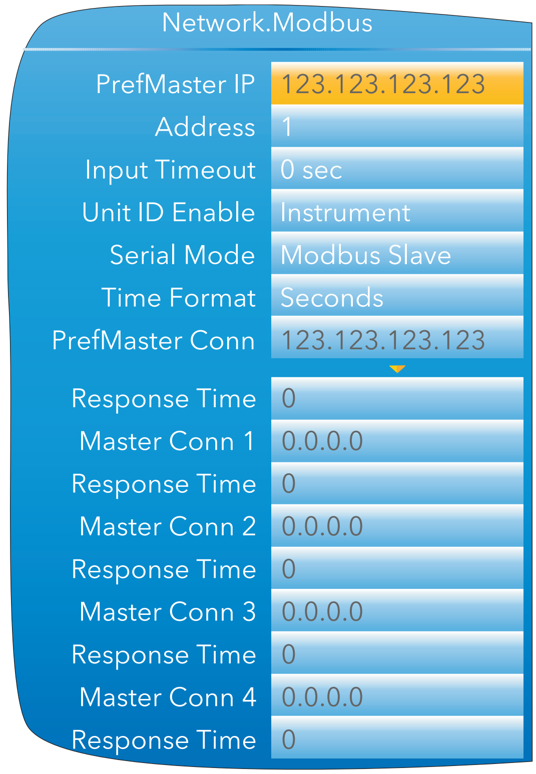



Figure 4.2.4 Modbus TCP configuration menu

| | |
|---|---|
| PrefMaster IP | The IP address of the relevant Modbus master. The Preferred master is guaranteed to be able to connect, even if all slave connections (max. = 4 for TCP) are in use. |
| Address | The Modbus address for this slave. This address must be unique for the network to which it is attached. The recorder will respond to this address and to Address 255. |
| Input Timeout | Allows a value of between 0 and 3600 seconds to be entered to set the timeout period for modbus input channels. If a modbus input is not written to within this period the value of the channel is set to -9999.0 with a 'No Data' status. A value of 0 disables the comms inactivity timeout feature. |
| Unit ID Enable | Enables/Disables the checking of the Modbus TCP unit identity field. |
| | Strict: The Modbus TCP Unit Identity Field (UIF) does not have to match the instrument address. The instrument responds only to Hex value FF in the UIF. iTools finds this instrument only at location 255, and then stops scanning. |
| | Loose: The Modbus TCP Unit Identity Field (UIF) does not have to match the instrument address. The instrument responds to any value in the UIF |
| | Instrument: The Modbus TCP Unit Identity Field (UIF) must match the instrument address or no response will be made to messages. |
| Serial Mode | Slave communications via the side mounted configuration port interface (CPI) clip (for iTools use.) Parameters: Baud rate 19,200; Parity = none; Number of data bits = 8; Number of stop bits = 1; no flow control. Can be set to 'Modbus Slave' or 'Off'. The unit must be restarted before any change takes effect. |
| Time Format | Allows the user to choose milliseconds, seconds, minutes or hours as the time format. Sets the resolution for the reading and writing of time format parameters. |
| PrefMaster Conn | Read only. Shows the IP address of the preferred master, when connected. |
| Response Time | Read only. Shows the response time for a single communications request to the relevant master. |
| Master Conn 1 to 4 | Read only. Shows the IP addresses of any other masters connected to this recorder. |

## 4.3 GROUP CONFIGURATION

Group configuration is separated into two areas, one which defines trending characteristics (for display channels) the other defining the recording characteristics for saving data to the Flash memory ready for archiving.

### 4.3.1 Group Trend configuration

This allows the user to define which points are to be traced on the display and at what interval, and also allows the number of chart divisions to be set up. Figure 4.3.1 shows a typical configuration page.

Note: The background chart colour is set up as a part of Instrument Display configuration (section 4.1.3)

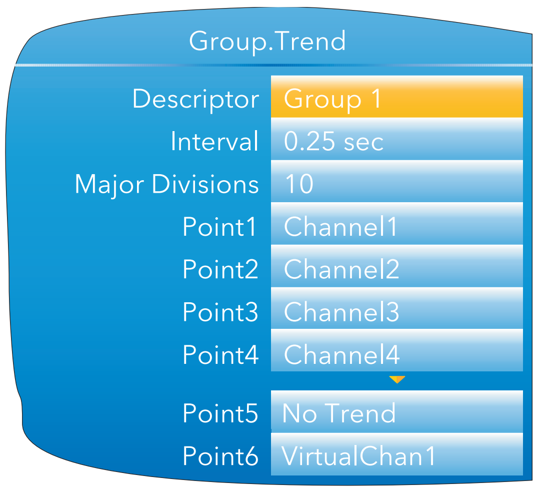



Figure 4.3.1 Group Trend Configuration

| | |
|---|---|
| Descriptor | Allows the user to enter a descriptor (20 characters max.) for the group. |
| Interval | The trending interval which defines how much data appears on one screen height or width. A number of discrete intervals can be chosen between 0.125 seconds to 1 hour. The selection should be made according to how much detail is required, and how much data is to be visible on the screen. |
| Major Divisions | Allows the user to select the number of divisions into which the scale is divided and how many gridlines are displayed. Setting the value to 1 results in just the zero and full scale values appearing. Setting the value to 10 (the maximum) results in a scale with zero, full scale and nine intermediate values appearing, with associated grid lines. |
| Point1 to Point6 | Allows the user to select which channels and virtual channels are to be traced. The maximum number of traces is six. |

### 4.3.2 Group Recording configuration

Similar to Trend configuration, above, but for saving the data to Flash memory history files. Each point can individually be enabled or disabled for recording, or recording can be disabled for the whole group.

Figure 4.3.2 shows a typical page.

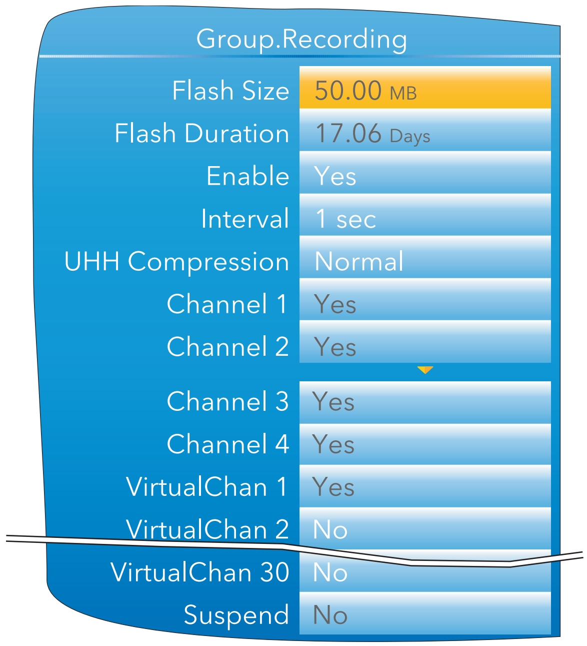



Figure 4.3.2 Group trend recording configuration

| | |
|---|---|
| Flash Size | Read only. Shows the size of the Flash memory fitted in MB. |
| Flash Duration | Read only. Shows the time it will take to fill the Flash memory if the recorder configuration remains unchanged. |
| Enable | 'Yes' enables group recording so that all points set to 'Yes' are stored in the recorder's flash memory. 'No' disables group recording. |
| Interval | Defines the rate at which data is saved to the recorder's Flash memory. The value affects how much trace history appears on the screen in trend history mode. |
| UHH Compression | Select 'Normal' or 'High'. 'Normal' compresses the data, but still provides an exact copy. 'High' compresses more, but values are saved only to 1 part in $10^8$ resolution. See also note 1, below. |

Channel 1 to VirtualChan 30 (see note 2, below)

Read only (greyed 'yes') for points being trended, (these are automatically recorded). For non-trending points the user may enable or disable each point individually.

| | |
|---|---|
| Suspend | Ignored unless the user has wired to this field. If wired then when set to 'No' recording is active, when set to 'Yes' recording is paused. |

Notes:

1. Where very high values are involved, such as in some totaliser values, 'High' compression may cause the value displayed at the recorder, and held in the history file, to be incorrect. The problem may be resolved by changing to 'Normal' compression, or, in the case of a totaliser, by re-scaling it (for example from MegaWatt hours to TeraWatt hours).
2. Virtual channels 1 to 15 are included in the standard build. Channels 16 to 30 are included only if the Modbus Master and / or EtherNet/IP option is fitted.

## 4.4 INPUT CHANNEL CONFIGURATION

Channel

N Select channel number

| Main | Trend | Alarm 1/2 |
|---|---|---|
| (Section 4.4.1) | (Section 4.4.2) | (Section 4.4.3) |
| Descriptor | Colour | Type |
| Type | Span low | Status |
| PV | Span high | Threshold |
| Status | | Hysteresis |
| IP adjust state | | Latch |
| Resolution | | Block |
| Units | | Dwell |
| Test signal | | Acknowledge |
| Input low/high | | Active |
| Shunt value | | Inactive |
| Linearisation type | | Not acknowledged |
| Range low/high | | Acknowledgement |
| Scale low/high | | |
| Offset | | |
| CJ type | | |
| Filter | | |
| Sensor break type | | |
| Fault Response | | |
| Sensor break value | | |
| Measured value | | |
| Internal CJ Temp | | |

Figure 4.4  Channel configuration menu

### 4.4.1 Channel Main

This section describes all possible menu items, but it should be noted that some items are context dependent (e.g. Cold Junction settings appear only for Type = 'Thermocouple').

Channels one to four in the configuration relate to An In 1 (terminals 1I, 1+ and 1-) to An In 4 (terminals 4I, 4+ and 4-) respectively - see figure 2.2.

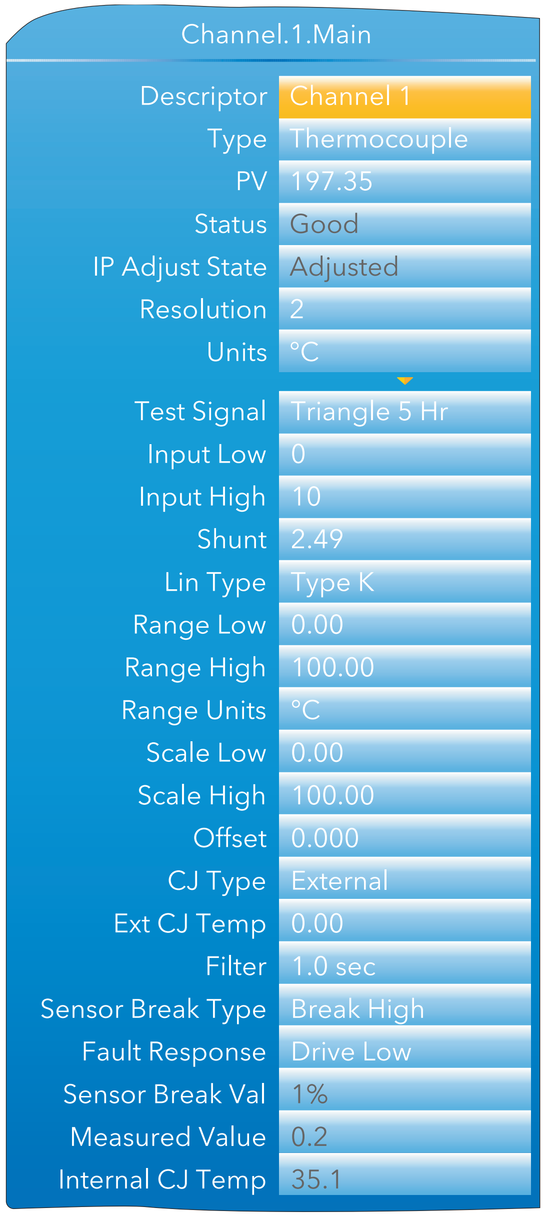



Figure 4.4.1a Channel main menu (expanded)

Note: For the sake of completeness, the figure above shows all possible fields, even though many are mutually exclusive. For example, 'Test signal' appears only when 'Test' is selected as Type. It would never appear when Type = thermocouple (as shown). Similarly, 'Shunt' would appear only for Type = mA.

### 4.4.1 CHANNEL MAIN (Cont.)

| | |
|---|---|
| Descriptor | Allows a (20 character max.) descriptor to be entered for the channel. Some thought should be given to ensure that the descriptor is meaningful because in some display screens it is truncated. For example, 'Furnace 1 area 1' and 'Furnace 1 area 2' might both appear as 'Furnace 1 a' and thus be indistinguishable from one another, except in background colour. |
| PV | Read only. Displays the current value of the channel. |
| Status | Read only. Shows the channel status as one of: 'Good', 'Channel Off', 'Over range', 'Under range', 'HW error', 'Ranging', 'HW (capability) exceeded'. |
| PV2 | Read only. For dual inputs only, displays the current value of the secondary input. |
| Status2 | Read only. For dual inputs only, shows the secondary input status (as 'Status' above). |
| IP Adjust State | Appears only for channels which have been included in the 'Adjust Input' procedure described in section 4.1.9. |
| IP Adjust State2 | As 'IP Adjust State', above but for secondary channels. |
| Resolution | Allows the number of decimal places to be defined for the channel. Valid entries are zero to nine. |
| Units | Allows a units string of up to five characters to be entered. |
| Type | Allows the user to select an input type for the channel. Available selections are: 'Off', 'Thermocouple', 'mV', 'V', 'mA', 'RTD', 'Digital', 'Test' or 'Ohms'. If the Dual Input option is fitted, Dual mV, Dual mA, Dual T/C (if enabled) are also available. |

Note : If Dual T/C is selected then it is essential that the secondary T/C input is field calibrated using the Input Adjust procedure (section 4.1.9)

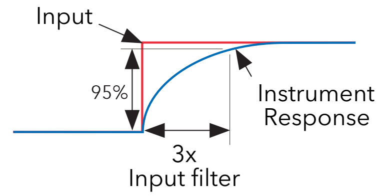
| | |
|---|---|
| Test signal | Appears only if 'Test' is selected as 'Type'. Allows either a sinusoidal or a triangular waveform to be selected at one of a number of cycle times between 40 seconds and five hours. |
| Input Low* | For Type = mV, Dual mV, V, mA, Dual mA or Ohms, the lowest value of the applied signal in electrical units. |
| Input High* | As 'Input Low', but the highest value of the applied signal in electrical units. |
| Shunt value | For mA and Dual mA input types only, this allows the value of the shunt resistor (in Ohms) to be entered. The recorder does not validate this value - it is up to the user to ensure that the value entered here matches that of the shunt resistor(s) fitted. For Dual mA input type, both primary and secondary inputs must have independent shunts each of the same value. |
| Lin type | Linear, Square root, x3/2, x5/2, User Lin.<br>Thermocouple types (alphabetical order): B, C, D, E, G2, J, K, L, N, R, S, T, U, NiMo/NiCo, Platinel, Ni/MiMo, Pt20%Rh/Pt40%Rh.<br>User 1 to User 4<br>Resistance thermometer types: Cu10, Pt100, Pt100A, JPT100, Ni100, Ni120, Cu53.<br>See Appendix A for input ranges, accuracies etc. associated with the above thermocouple and RTD types. See section 4.13 for details of user linearisations. |
| Range Low* | For thermocouples, RTDs, User linearisations and retransmitted signals only, the lowest value of the required linearisation range. |
| Range High* | For thermocouples, RTDs, User linearisations and retransmitted signals only, the highest value of the required linearisation range. |
| Range Units | For thermocouples only and RTDs, Select °C, °F or K. |
| Scale Low/High | Maps the process value to (Scale High - Scale Low). For example, an input of 4 to 20mA may be scaled as 0 to 100% by setting Scale low to 0 and Scale High to 100. |
| Scale Low2/High2 | As 'Scale Low/High but for the secondary input (PV2). |
| Offset | Allows a fixed value to be added to or subtracted from the process variable. |

*Note: See section 4.13 for details of the configuration of Range High/Low and Input High/Low when 'Type' = User 1 to User 4.

### 4.4.1 CHANNEL MAIN (Cont.)

Offset2 — The nature of the secondary input results in an offset being introduced into the process variable value.
For mA inputs this offset is removed automatically, without user intervention.
For mV inputs the offset depends on the value of the voltage source impedance and is equal to 199.9μV/Ω. This offset can be compensated for either by using this Offset2 parameter, or by carrying out the 'Input Adjust' procedure (Section 4.1.9).
For Dual T/C inputs, it is recommended that the 'Input Adjust' procedure be used instead of Offset2 as the use of Offset2 results in an offset which is non-linear over the thermocouple range.

Input filter — Damping can be used to filter out noise from slowly changing signals so that the underlying trend can be seen more clearly. Valid input values are between 0 and 60 seconds.




Note: Applying a filter to an input channel can affect the operation of any Rate-of-change alarms configured to act on that channel.

CJC Type — For thermocouple input types only, this allows the user to select 'None', 'Internal', 'External' or 'Remote 1' to 'Remote 4'. For Dual T/C inputs, both primary and secondary inputs use the same cold junction.
None: No Cold junction compensation applied.
'Internal' uses the recorder's internal cold junction temperature measurement.
'External' means that the cold junction is to be maintained by the user, at a fixed, known temperature. This temperature is entered in the 'External CJ Temp' field which appears when 'External' is selected.
Remote 1 (2) (3) (4) means that the cold junction temperature is being measured by input channel 1 (2) (3) (4) respectively. (This must be a different channel from that currently being configured).

Ext. CJ Temp — Appears only if CJC type is set to 'External', and allows the user to enter the temperature at which the external cold junction is being maintained.

Sensor Break Type — Defines whether the sensor break becomes active for circuit impedances greater than expected.
'Off' disables Sensor Break detection.
Break Low: Sensor break active if measured impedance is greater than the 'Break Low impedance' value given in table 4.4.1.
Break High: Sensor break active if measured impedance is greater than the 'Break High Impedance' value given in table 4.4.1.

For mA inputs, limits are applied, such that if the process value lies outside these limits, a sensor break is assumed to have occurred. These limits are (Input lo - 4% Span) and (Input high + 6% Span). For example, for a 4 to 20mA signal, an input below 3.36mA or above 20.96mA will trigger a sensor break event

| Range | Break Low impedance | Break High Impedance |
|---|---|---|
| 40mV | ~5kΩ | ~20kΩ |
| 80mV | ~5kΩ | ~20kΩ |
| 2V | ~12.5kΩ | ~70kΩ |
| 10V | ~12.5kΩ | ~120kΩ |

Table 4.4.1 Minimum impedances for sensor break detection

Note: Break High impedance values would be used typically for sensors which have a high nominal impedance when working normally

### 4.4.1 CHANNEL MAIN (Cont.)

Sensor Break type (Cont.)

Input sensor break detection is not supported for secondary inputs. The internal circuit acts as a 'pull up' on the secondary input which therefore saturates high in the event of a sensor break.

| | |
|---|---|
| Fault Response | Specifies the behaviour of the recorder if a sensor break is detected or if the input is over driven (saturated high or low).<br>'None' means that the input drifts, with the wiring acting as an aerial.<br>'Drive High' means that the trace moves to (Scale High +10%). 'Drive Low' means that the trace moves to (Scale Low -10%), where the 10% values represent 10% of (Scale High - Scale Low). |
| Sensor Break Val | A diagnostic representation of how close the sensor break detection circuitry is to tripping.<br>Measured ValueThe (read only) input channel measured value before any scaling or linearisation is applied. |
| Measured Value2 | As 'Measured Value', above but for the secondary input. |
| Internal CJ temp | The (read only) temperature of the internal cold junction associated with this channel. |

### 4.4.2 Channel Trend configuration

This area allows the configuration of channel colour and span.

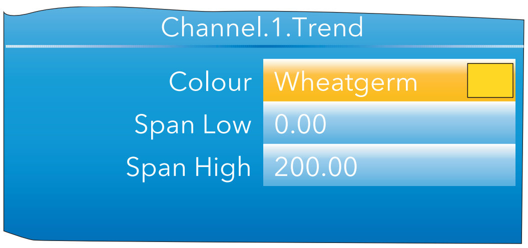



Figure 4.4.2a Channel Trend menu

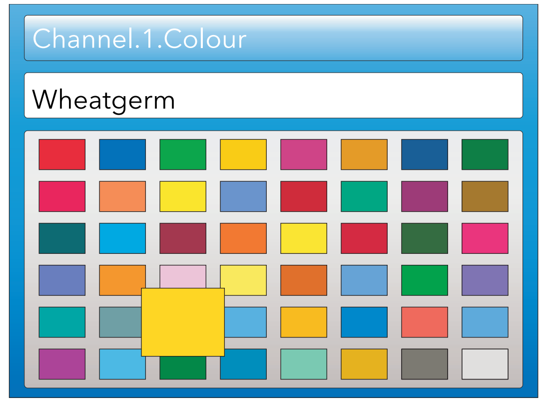



Figure 4.4.2b Colour selection

| | |
|---|---|
| Colour | Allows a colour to be specified for the channel. The Scroll key is used to enter the colour swatch page. The up and down arrows are used to scroll through the available colours, with each colour being enlarged for as long as it is 'selected'. Once the required colour, is reached, the scroll key is used again to return to the Trend Configuration. |
| Span Low/High | Span low and high values. |

Note: Trend colours and alarm settings for secondary inputs are configured in the maths channels to which they are wired.

**SPAN EXAMPLE**

In an input range of 0 to 600 degrees C, the temperature range between 500 and 600 degrees is of most interest. In such a case, Span Low is set to 500 and Span High to 600 so that the recorder trends only the required part of the temperature range, effectively magnifying the area of interest.

Note: Trending is restricted to the PV range (Span High - Span Low), but the instrument can display values outside this range.

**CHANNEL CONFIGURATION EXAMPLE**

A type J thermocouple is used to measure a temperature range of 100 to 200 degrees Celsius. This thermocouple output is transmitted to the recorder by a 4 to 20mA transmitter, for display as a value between 0 and 100%.

In Channel.Main, set the following for the relevant channel:

| | |
|---|---|
| Type | = mA |
| Units | = % |
| Input Low | = 4.00 |
| Input high | = 20.00 |
| Shunt | = 250 Ohms |
| Lin Type | = Type J |
| Range Low | = 100.00 |
| Range High | = 200.00 |
| Range Units | = °C |
| Scale Low | = 0 |
| Scale High | = 100 |

Other items may be left at their defaults.

### 4.4.3 Alarm 1 menu

Allows the alarm characteristics for Alarm 1 to be configured. The figure below shows a typical configuration page (expanded for clarity). Actual configuration parameters are context sensitive.

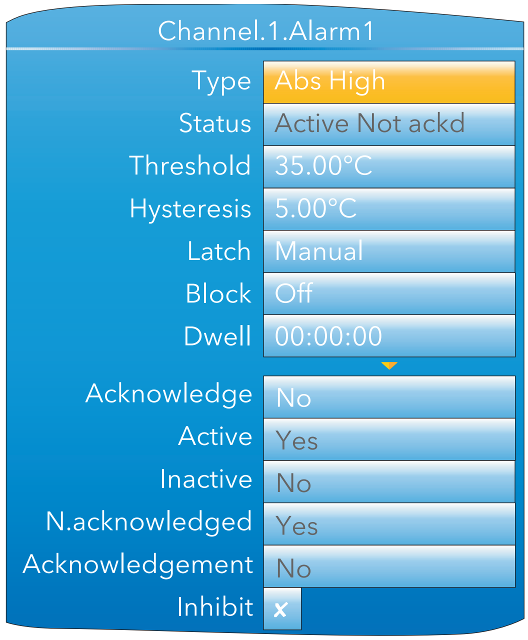



Figure 4.4.3 Typical alarm 1 configuration menu

| | |
|---|---|
| Type | Select an alarm type from: 'Off', 'Abs. High' (absolute high). 'Abs. Low' (absolute low), 'Dev. High' (deviation high), 'Dev. Low' (deviation low), 'Dev. Band' (deviation band), 'Rise ROC' (rate-of-change: rising), 'Fall ROC' (rate-of-change: falling), 'Digital High', 'Digital Low'. See 'Alarm types', below, for definitions. |
| Status | Read only. This shows that the alarm is Off, Active, SafeNotAcked or ActiveNotAcked. For 'Auto' and 'Manual' alarms only, 'SafeNotAcked' means that the alarm trigger source has returned to a non-alarm state, but the alarm is still active because it has not been acknowledged. Similarly, 'ActiveNotAcked' means that the source is still active and the alarm has not been acknowledged. Always shows 'Off' when the alarm is inhibited (see below). |
| Threshold | For absolute alarms only, this is the trip point for the alarm. For absolute high alarms, if the threshold value is exceeded by the process value (PV) of this channel, then the alarm becomes active, and remains active until the PV falls below the value (threshold - hysteresis). For absolute low alarms, if the PV of this channel falls below the threshold value, then the alarm becomes active and remains active until the PV rises above (Threshold + Hysteresis). |
| Reference | For deviation alarms only, this provides a 'centre point' for the deviation band.<br>For 'deviation high' alarms, the alarm becomes active if the process value (PV) rises above the value (Reference + Deviation) and remains active until the PV falls below (Reference + Deviation - Hysteresis).<br>For 'deviation low' alarms, the alarm becomes active if the process value (PV) falls below the value (Reference - Deviation) and remains active until the PV rises above (Reference - Deviation + Hysteresis).<br>For 'deviation band' alarms, the alarm is active whenever the process value (PV) lies outside the value (Reference ± Deviation) and remains active until the PV returns to within the band, minus or plus Hysteresis as appropriate. |
| Deviation | For deviation alarms only, 'Deviation' defines the width of the deviation band, each side of the Reference value, as described immediately above. |

### 4.4.3 ALARM 1 MENU (Cont.)

| | |
|---|---|
| Hysteresis | For absolute and deviation alarms, this provides a means of preventing multiple alarm triggering, if the process value is drifting close to the trigger value. |
| Amount | For rate-of-change alarms only. The alarm becomes active if the process value rises (Rise ROC) or falls (Fall ROC) by more than the specified 'Amount' within the time period defined in 'Change Time', below. The alarm remains active until the rate of change falls below the value (Amount/Change Time) in the relevant sense. |
| Change Time | Settable to 1 second, 1 minute or 1 hour. See 'Amount' (above). |
| Average Time | For rate-of-change alarms only. This allows an averaging period (for the process value) to be entered to reduce nuisance trips due to signal noise, or if the rate of change is hovering around the trip value. |
| Latch | None: the alarm remains active until the monitored value has returned to a non alarm state, when it becomes inactive.<br>Auto: The alarm remains active until the monitored value has returned to a non alarm state and the alarm has been acknowledged. Acknowledgement can take place either before or after the value has returned a non alarm state.<br>Manual: The alarm remains active until the monitored value has returned to a non alarm state and the alarm has been acknowledged. Acknowledgement is permitted only after the value has returned a non alarm state.<br>Trigger: Not enunciated, this mode is used only to initiate an action defined by user wiring either using iTools or using the user interface. |
| Block | Alarms with 'Block' set to 'On' are inhibited until the monitored value has entered the 'safe' condition after a start-up. This prevents such alarms from becoming active whilst the process is brought into control. If a latching alarm is not acknowledged then the alarm is re-asserted (not blocked), unless the alarm's threshold or reference value is changed, in which case the alarm is blocked again. |
| Dwell | Initiates a delay between the trigger source becoming active, and the alarm becoming active. If the trigger source returns to a non alarm state before the dwell time has elapsed, then the alarm is not triggered and the dwell timer is reset. |
| Acknowledge | Select 'yes' to acknowledge the alarm. Display returns to 'No'. |
| Active | Read only. Shows the status of the alarm as 'Yes' if it is active, or No, if inactive. The active/inactive state depends on the Latch type (above) and acknowledgment status of the alarm. Always shows 'No' if the alarm is inbited (below). |
| Inactive | As for 'Active' above, but shows 'Yes' if the alarm in inactive and 'No' if the alarm is active. Always shows 'Yes' if the alarm is inbited (below). |
| N.acknowledged | As for 'Active' above but shows 'Yes' for as long as the alarm is unacknowledged, and 'No' as soon as it is acknowledged. Always shows 'No' if the alarm is inbited (below). |
| Acknowledgement | Fleetingly goes 'Yes' on alarm acknowledgement, and then returns to 'No'. |
| Inhibit | When 'Inhibit' is enabled, (tick symbol), the alarm is inhibited. Status is set to 'Off'; 'Active' and 'N.acknowledged' are set to 'No', and 'Inactive' is set to 'Yes'. If the alarm is active when inhibit is enabled, then it becomes inactive until inhibit is disabled, when its status depends on its configuration. Similarly if the alarm trigger becomes active when the alarm is inhibited, the alarm remains 'off' until inhibit is disabled, when its status depends on its configuration. |

### 4.4.4 Alarm 2 menu

As above for Alarm 1 menu.

Note: The parameters 'Acknowledge', 'Active', 'Inactive', 'N(ot) Acknowledged' and, 'Acknowledgement' can all be 'wired' to other parameters, so, for example, a relay can be made to operate whilst the alarm is inactive or whilst it is active or on acknowledgement etc. by wiring the relevant parameter to the relay's 'PV' input. See section 7 for details of user wiring.

### 4.4.5 Alarm types

The following figures attempt to show graphically the meanings of the alarm parameters which can be set for the various alarm types available.

#### ABSOLUTE ALARMS

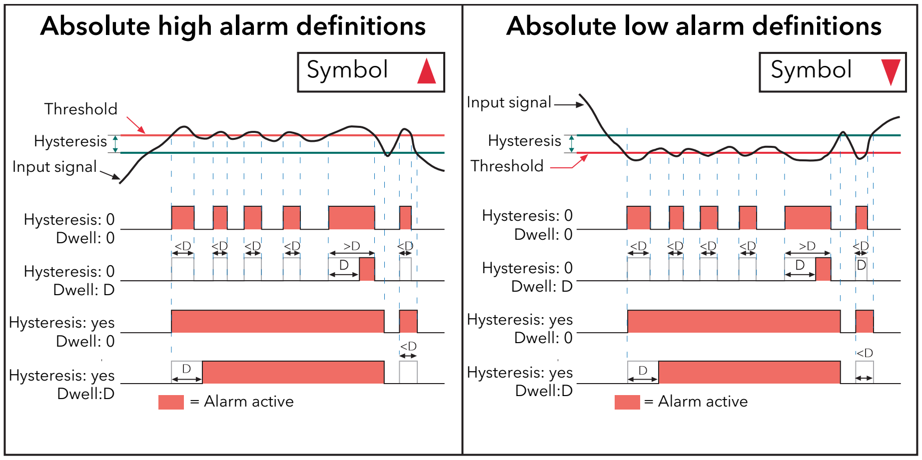



Figure 4.4.5a absolute alarm parameters

#### DEVIATION ALARMS

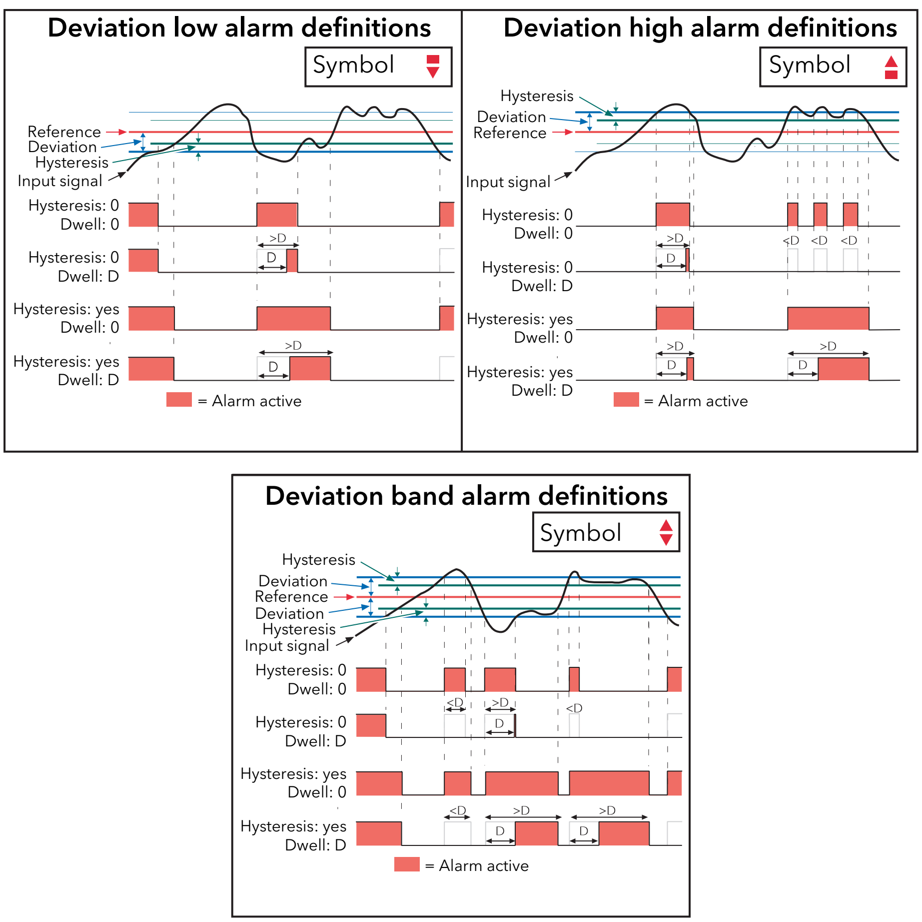
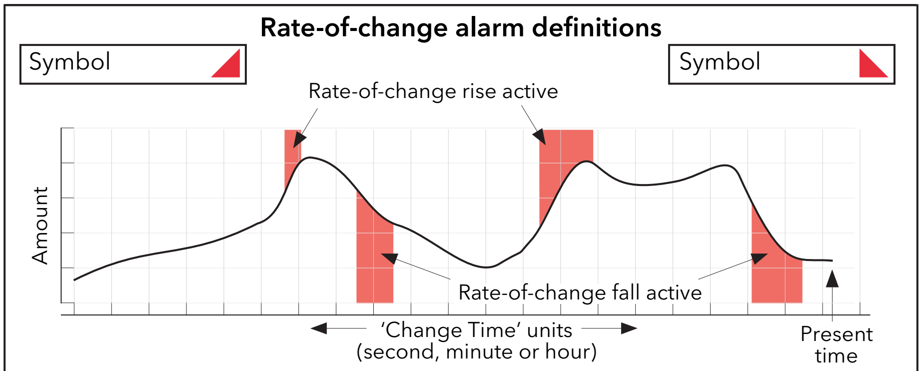



Figure 4.4.5b Deviation alarm parameters

### 4.4.5 ALARM TYPES (Cont.)

#### RATE-OF-CHANGE ALARMS




A rate-of-change alarm is active if the monitored value changes by more than one 'Amount' in less than 1 'Change time'. Dwell delays the alarm on-time. Averaging removes the effects of signal noise.

Figure 4.4.5c Rate-of-change alarm parameters

Note: Operation of rate-of-change alarms may be affected if an input filter (section 4.4.1) is applied to the input signal.

## 4.5 VIRTUAL CHANNEL CONFIGURATION

This allows the configuration of maths channels, totalisers and counters. The configuration is divided into the following areas: 'Main', 'Trend', 'Alarm 1*' and 'Alarm 2*'. Items appearing in the 'Trend', Alarm 1' and 'Alarm 2' areas are identical with the equivalent items described in section 4.4 (Input channels), above.

*Note: Virtual channels 16 to 30 (supplied with Modbus Master and EtherNet/IP options only) come without alarms.

### 4.5.1 Maths channel configuration

The following maths functions are available (listed in up-arrow scroll order)

Off, Add, Subtract, Multiply, Divide, Group Average, Group minimum, Group maximum, Modbus input, Copy, Group minimum (latch), Group maximum (latch), Channel maximum, Channel minimum, Channel Average, Configuration revision, Off.

Figure 4.5.1 shows a typical maths channel configuration

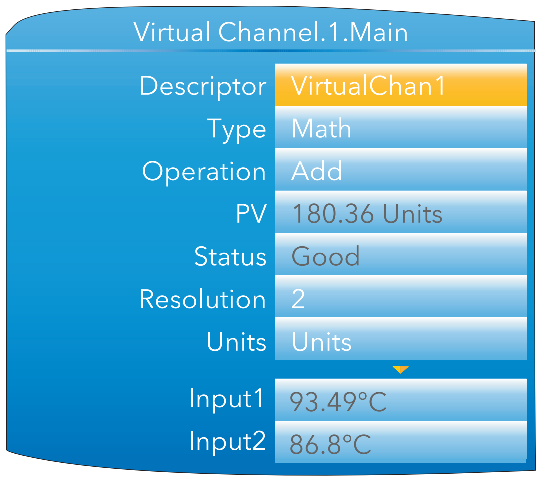



Figure 4.5.1 Maths channel configuration (typical)

| | |
|---|---|
| Descriptor | Allows the user to enter a descriptor (20 characters max.) for the maths channel |
| Type | Math selected for this example. (See sections 4.5.2 and 4.5.3 for totalisers and counters respectively.) |
| Operation | Allows the user to select the required maths function. See 'Maths Functions', below. |
| PV | Read only. Shows the dynamic value of this channel in the units entered in 'Units' below. |
| Status | Read only. Shows the status of this channel, reflecting the status of the input sources. |
| Resolution | Enter the number of decimal places required |
| Units | Allows a five character string to be entered to be used as the channel units. |
| Input1 | The value of input 1. May be entered manually, or it may be wired from another parameter (section 7). Uses the resolution of the source. |
| Input 2 | As for 'Input 1', Appears only when the operation requires two inputs. |
| Reset | Allows the user to reset latching functions (e.g. Channel Max) or averaging functions (e.g. Channel Avg). Reset is carried out by setting the field to 'Yes', then operating the scroll key. The display returns to 'No'. Alternatively the function can be reset by another parameter wired to 'Reset'. |
| Time Remaining | The period of time remaining before the virtual channel performs its operation. For example, the time remaining for the maths channel average operation to sample the input before performing the calculation. |
| Period | For averaging functions, this allows a period to be entered, over which the value is to be averaged. Selectable periods are: 0.125, 0.25, 0.5, 1, 2, 5, 10, 20, 30 seconds, 1, 2, 5, 10, 20, 30 minutes, 1, 2, 6, 12, 24 hours |

### 4.5.1 MATHS CHANNEL CONFIGURATION (Cont.)

#### MATHS FUNCTIONS

| | |
|---|---|
| Off | Out = -9999; status = Off |
| Add | Out = Input1 + Input2 |
| Subtract | Out = Input1 - Input2 |
| Multiply | Out = Input1 x Input2 |
| Divide | Out = Input1 ÷ Input2. If Input2 = 0, Out = -9999; Status = 'Bad'. |
| Group Avg* | Out = Instantaneous sum of all points in the recording group (except this one and any channel that has been configured with operation = group average, group minimum, group maximum, group minimum (latched), group maximum (latched), channel maximum or channel minimum), divided by the number of points in the group (excluding this one).<br>Any point that has a status other than 'Good' is excluded from the calculation.<br>If the group contains no channels, Out = -9999; Status = 'No data'. |
| Group Min* | Out = Instantaneous value of whichever point (except this one) in the recording group has the lowest value.<br>Any point that has a status other than 'Good' is excluded from the calculation.<br>If the group contains no channels, Out = -9999; Status = 'No data'. |
| Group Max* | Out = Instantaneous value of whichever point (except this one) in the recording group has the highest value.<br>Any point that has a status other than 'Good' is excluded from the calculation.<br>If the group contains no channels, Out = -9999; Status = 'No data'. |
| Modbus Input | Out = value written to this channel's modbus input.<br>If the comms timeout expires, Out = -9999; status = 'No data'. |
| Copy | Allows an input or other derived channel to be copied. |
| Grp Min Latch* | Out = Lowest value reached by any point in the recording group (except this one) since last reset.<br>Any point that has a status other than 'Good' is excluded from the calculation.<br>If the group contains no channels, Out = -9999; Status = 'No data'. |
| Grp Max Latch* | Out = Highest value reached by any point in the recording group (except this one) since last reset.<br>Any point that has a status other than 'Good' is excluded from the calculation.<br>If the group contains no channels, Out = -9999; Status = 'No data'. |
| Channel Max | Out = Highest value reached by Input1 since last reset.<br>If Input1 has a status other than 'Good', then Out = -9999 and 'Status' depends on the status of Input1. |
| Channel Min | Out = Lowest value reached by Input1 since last reset.<br>If Input1 has a status other than 'Good', then Out = -9999 and 'Status' depends on the status of Input1. |
| Channel Avg | Out = the average value of Input1 over the time specified in 'Period'.<br>If Input1 has a status other than 'Good', then Out = -9999 and 'Status' depends on the status of Input1. |
| Config Revision | Out = current Configuration Revision value. |

*Note: All 'Group' functions operate on the 'Recording' group, not on the 'Trend' group.

### 4.5.2 Totaliser configuration

Totalisers allow the user to maintain a running total of any input channel, or of any maths channel. Using maths channels, it is possible to totalise combinations of input channels so that, for example, the sum of two channels or the difference between them could be totalised if required.

The Rollover Value of the totaliser is configurable (default 1,000,000). At the point of rollover, the Rollover output is set. This output can be used to expand the range of the totaliser by wiring it to the Trigger of a counter.

Wiring is carried out either at the operator interface (section 7) or in iTools (section 6).

The totaliser equation is:

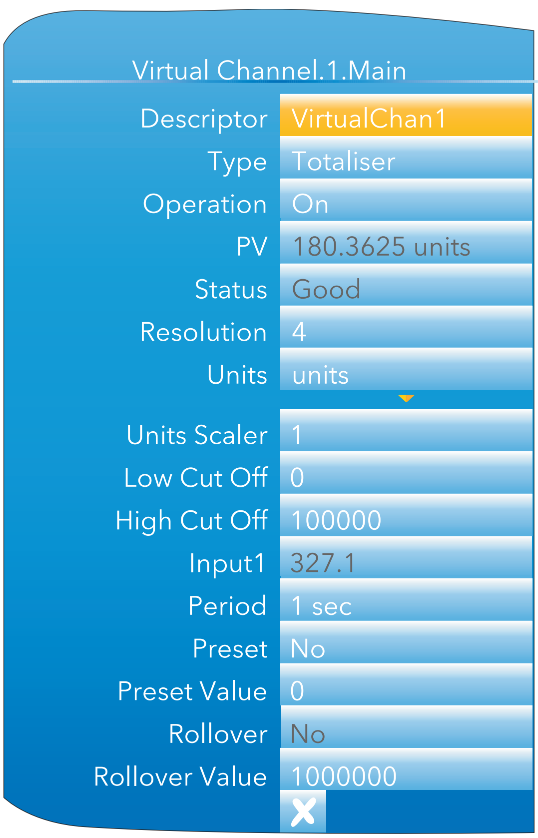
$$tot_t = tot_{t-1} + \frac{ma_t}{PSF \times USF}$$

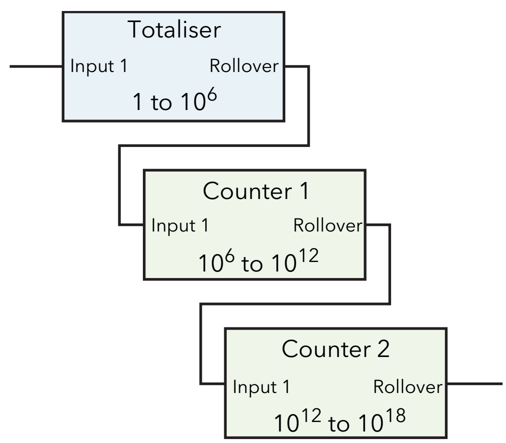
where,

$tot_t$ = totaliser value this sample
$tot_{t-1}$ = totaliser value last sample
$ma_t$ = process value this sample
PSF = Period Scaling Factor (Period)
USF = Units Scaling Factor (Units scaler)

Note: the time between samples is 125ms.

Figure 4.5.2 shows a typical configuration page.




Using cascaded counters to expand the totalisation range.

Figure 4.5.2 Typical totaliser configuration menu

| | |
|---|---|
| Descriptor | Allows the user to enter a descriptor (20 characters max.) for the totaliser. |
| Type | Select: Math, Counter or Totaliser. |
| Operation | Allows the user to enable ('On') or disable ('Off') the totaliser. |

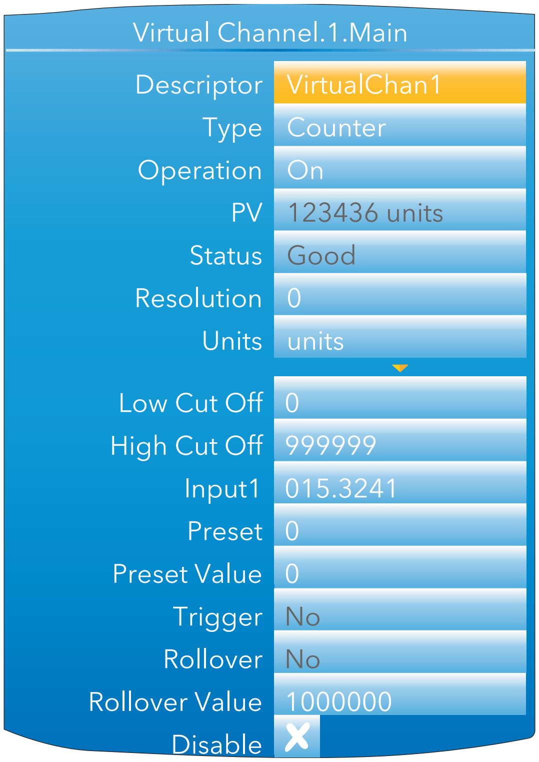
### 4.5.2 TOTALISER CONFIGURATION (Cont.)

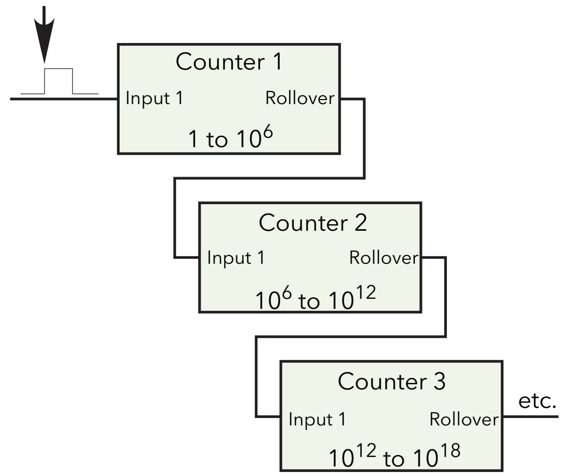
| | |
|---|---|
| PV | Read only. Shows the dynamic value of the totaliser. |
| Status | Read only. Shows the status of the totaliser. |
| Resolution | Allows the number of decimal places (up to 6) to be selected for the totaliser.UnitsAllows a units string of up to five characters to be entered for the totalised value. |
| Units Scaler | Allows a units scaler to be selected. If, for example, the input channel has units of litres per hour, then, if the Units Scaler is set to one, the totalised value will be in litres. If the Units Scaler is set to 1000, then the totalised value will be in thousands of litres. Setting the Units Scaler to a negative value, causes the totaliser to decrement rather than increment. |
| Low Cut Off | Used to restrict the input operating range of the totaliser. Minimum value = -100 000 |
| High Cut Off | Used to restrict the input operating range of the totaliser. Maximum value = 100 000 |
| Input1 | The value of the source. May be entered manually, or this parameter can be wired from an external channel PV. |
| Period | The totaliser equation works in seconds. If the totalised channel units are other than 'per second', a period scaler different from the default (1 sec) must be used. The 'Period' field presents a number of fixed periods from 0.125 seconds to 24 hours for selection. |
| Preset | Setting this to 'Yes' causes the totaliser to adopt the Preset Value. The field returns immediately to 'No'. The totaliser can also be preset by an external source 'wired' to this parameter. |
| Preset Value | Allows the entry of a value, from which the totaliser is to start incrementing or decrementing. The direction of the count is set by the sign of the units scaler: positive = increment; negative = decrement. |
| Rollover | This is the rollover output which will be set for one execution cycle when the totaliser rolls over. This output can be used to expand the range of the totaliser by wiring it to the Trigger input of a counter. |
| Rollover Value | This is the value at which the totaliser will rollover. When the totaliser rolls over the difference between the rollover value and the calculated output will be added to 0.<br>Example 1: with a rollover value of 1000 and a current output of 999 and an input of 5, then the output will become 4.<br>Example 2: with a rollover value of -1000 and a current output of -999 and an input of -5, then the output will become -4.<br>Note: in both examples, the Rollover output will be set for 1 execution cycle. |
| Disable | Allows the user temporarily to suspend totalising action. The output retains the pre-disabled value until the totaliser is re-enabled, when it resumes from that value. The totaliser is toggled between being enabled (cross symbol) and disabled (tick symbol) by means of the scroll key. |

### 4.5.3 Counter configuration

This allows the user to set up a counter to count trigger inputs (or it may be incremented from the Configuration page. The Rollover Value of the counter is configurable (default 1,000,000). Counters can be cascaded by wiring from 'Rollover' of one counter to 'trigger' of the next. Wiring is carried out from the operator interface (section 7) or in iTools (section 6).

For 'Trend', 'Alarm 1' and 'Alarm 2' configurations please see the relevant parts of section 4.4.



Cascading counters

Figure 4.5.3 Typical Counter configuration

| | |
|---|---|
| Descriptor | Allows the user to enter a descriptor (20 characters max.) for the counter. |
| Type | Select: Math, Counter or Totaliser. |
| Operation | Allows the user to enable ('On') or disable ('Off') the counter. |
| PV | Read only. Shows the dynamic value of the counter. |
| Status | Read only. Reflects the status of the input channel. |
| Resolution | Allows the number of decimal places (up to six) to be defined for the channel. |
| Units | Allows a units string of up to five characters to be entered for the counter value |
| Low Cut Off | Specifies a value below which the counter will not decrement. |
| High Cut Off | Specifies a value above which the counter will not increment. |
| Input1 | The amount by which the counter is incremented each time 'Trigger' goes high. The value may be entered manually, or wired from another parameter. Negative values cause the counter to decrement. |
| Preset | Setting this to 'Yes' causes the counter to adopt its Preset Value. The field returns immediately to 'No'. The counter can also be preset by wiring from another parameter. |
| Preset Val | Allows the entry of a value, from which the counter is to start incrementing or decrementing. |
| Trigger | Setting this to 1, causes the current value of the input source to be added to the Counter value. This function can be carried out manually, or the input can be wired from another parameter (section 7.2). |
| Rollover | This is the rollover output which will be set for one execution cycle when the counter rolls over. This output can be used to expand the range of the cascade counters by wiring it to the Trigger input of the next counter. |
| Rollover Value | This is the value at which the counter will rollover. When the counter rolls over the difference between the rollover value and the calculated output will be added to 0. Example 1: with a rollover value of 1000 and a current output of 999 and an input of 5, |

### 4.5.3 Counter configuration (Cont.)

then the output will become 4 when the counter is next triggered.
Example 2: with a rollover value of -1000 and a current output of -999 and an input of -5, then the output will become -4 when the counter is next triggered.
Note: in both examples, the Rollover output will be set for 1 execution cycle.

Disable — Allows the user temporarily to suspend counting. The output retains the pre-disabled value until the counter is re-enabled, when it resumes counting from that value. The counter is toggled between being enabled (cross symbol) and disabled (tick symbol) by means of the scroll key.

## 4.6 LOOP OPTION CONFIGURATION

This configuration area allows the user to set up two control loops. This description refers to temperature control loops, but the configuration parameters apply equally to other types of control. For each loop, channel 1 is assumed to be a heating channel; channel 2 a cooling channel.

The configuration is divided into a number of areas, as shown in the overview below.

Loop

N    N = 1 or 2

| Main | Setup | Tune | PID | Setpoint | Output | Diagnostics |
|---|---|---|---|---|---|---|
| (Section 4.6.1) | (Section 4.6.2) | (Section 4.6.3) | (Section 4.6.4) | (Section 4.6.5) | (Section 4.6.6) | (Section 4.6.7) |
| Auto/Manual | Loop name | Tune Enable | Sched Type | Range Low | Output Low | Error |
| PV | Ch1 Control | Low Output | No. of sets | Range High | Output High | Target Output |
| Inhibit | Ch2 Control | High Output | Remote i/p | SP Select | Ch1/2 Output | Working Out Low |
| Target Setpoint | Control action | State | Active set | SP1 | Ch2 Deadband | Working Out High |
| Working Set-point | PB Units | Stage Time | Boundary 1-2 | SP2 | Ch1/2 OnOff Hyst | Loop Break |
| Working Output | Deriv Type | | Boundary 2-3 | SP Low Limit | Ch1/2 Travel time | Prop Output |
| Integral Hold | Setpoint Access | | PB | SP High Limit | Nudge raise | Integral Output |
| | Auto/Man Access | | Ti | Alt SP Enable | Nudge lower | Deriv Output |
| | | | Td | Alt SP | Ch1/2 pot pos | Sensor Break |
| | | | CBH | Rate | Ch1/2 pot brk | Sched PB |
| | | | CBL | Rate Done | Pot Brk mode | Sched Ti |
| | | | MR | SP rate disable | Rate | Sched Td |
| | | | LBT | Servo to PV | SbrkMode | Sched CBH |
| | | | Output Low | SP Trim | Sbrk OP | Sched CBL |
| | | | Output High | SP Trim Low | Safe OP | Sched MR |
| | | | | SP Trim High | Manual Mode | Sched Loop Break |
| | | | | Manual Track | Manual Output | Sched Out Low |
| | | | | Setpoint Track | Forced Output | Sched Out High |
| | | | | Track PV | Manual Startup | |
| | | | | Track Value | Pff En | |
| | | | | SP Int Balance | Power In | |
| | | | | | Cool type | |
| | | | | | FF type/gain | |
| | | | | | FF offset/trim lin | |
| | | | | | FF OP | |
| | | | | | Track Output | |
| | | | | | Track Enable | |
| | | | | | Rem. Output Low | |
| | | | | | Rem. Output High | |

Figure 4.6 Loop configuration overview

For a general discussion of control loops, please see Appendix B to this manual.

### 4.6.1 Main menu parameters

| | |
|---|---|
| Auto/Manual | Selects Auto(matic) or Manual operation. 'Auto' automatically controls output power in a closed loop configuration. In manual mode, the operator controls the output power. |
| PV | The Process Variable input value. The value can be entered by the user, but is most often 'wired' from an analogue input. |
| Inhibit | Select 'No' or 'Yes'. 'Yes' stops the loop and sets the output to a 'safe' value, this value being entered as a part of the Output configuration (section 4.6.6). If an output rate limit is set, then the output ramps to the safe level at that rate, otherwise it performs a step change. If setpoint or manual tracking is enabled (in setpoint configuration section 4.6.5), Inhibit overrides tracking.<br>If 'No' is selected, the loop operates normally.<br>Inhibit can be enabled/disabled from an external source. |
| Target Setpoint | The value at which the control loop is aiming. SP may be derived from a number of sources, as described in Appendix B, section B2.5. The value range limited by the setpoint limits (SP High Limit and SP Low Limit) described in section 4.6.5. |
| Working Setpoint | A read-only value displaying the current value of setpoint being used by the loop. This might or might not be the Target setpoint. The value may come from a number of sources, but is limited by the setpoint limits (SP High Lim and SP Low Lim) described in section 4.6.5. |
| Working Output | The actual working output value before being split into channel 1 and 2 outputs. |
| Integral Hold | Select 'Yes' or 'No'. 'Yes' freezes the integral term at its current value. IntHold ensures that the power is reapplied smoothly after the loop has been broken for service reasons, for example. |

### 4.6.2 Setup menu parameters

| | |
|---|---|
| Loop Name | Allows entry of an 11 character name for the loop. |
| Ch1 Control | Selects the type of control for channel one from:<br>Off: Channel is turned off<br>OnOff: Channel uses on/off control<br>PID: Proportional + integral + derivative (three-term) control.<br>VPU: Valve positioning unbounded<br>VPB: Valve positioning bounded.<br>Appendix B, Section B2.2 provides more details. |
| Ch2 Control | As above, but for loop channel two. |
| Control Action | Select 'Reverse' or 'Direct'.<br>'Reverse' means that the output is 'on' when the process value (PV) is below the target setpoint (SP). This is normal for heating control.<br>'Direct' means that the output is on when PV is above SP. This is normal for cooling control. |
| PB Units * | Select 'Engineering' or 'Percent'.<br>'Engineering' displays values in (for example) temperature units (e.g. °C or °F).<br>'Percent' displays values as a percentage of loop span (Range Hi - Range Lo). |
| Deriv Type * | 'Error' means that changes to PV or SP cause changes to the derivative output. Derivative on error should be used with a programmer since it tends to reduce ramp overshoot. 'Error' provides rapid response to small setpoint changes which makes it ideal for temperature control systems.<br>'PV' means that changes in PV alone cause changes to the derivative output. Typically used for process systems using valve control, as it reduces wear on the valve mechanics. |
| Setpoint Access | Allows setpoint editing permission in the loop display pages (section 3.4.7).<br>'Read/Write' allows free access to all users<br>'Read Only' allows editing only in Configuration or Supervisor modes.<br>'Operator R/W' allows editing in all modes except 'Logged out'. |

**4.6.2 SETUP MENU PARAMETERS (Cont.)**

| | |
|---|---|
| Auto/Man Access | As 'Setpoint Access' above, but for Auto/manual parameter. |

*Note: 'PB Units' and 'Deriv Type' appear only if at least one of Ch1 Control and Ch2 Control is set to 'PID', 'VPU' or' VPB'.

### 4.6.3 Tune menu parameters

| | |
|---|---|
| Tune R2G | Defines the type of relative cooling gain tuning for the loop.<br>'Standard' - tunes the relative cooling gain of the loop using the standard R2G tuning algorithm.<br>'R2GPD' - If the process is heavily lagged, this setting should be used.<br>'Off' - R2G is not calculated automatically. Enter the value manually as described in section B2.4.7 Manual tuning.<br>'Manual Tuning'.<br>Note: This parameter only appears when both channel 1 and channel 2 are configured (for example, in heat/cool processes).<br>For futher information, refer to section B2.4.6 Relative Cool Gain in Well Lagged Processes. |
| Tune Enable | 'On' initiates autotune. Legend changes to 'Off' when autotune is complete. Can be set to 'Off' manually, to stop the tuning process. |
| Low Output | Sets a low limit to be imposed whilst autotune is running. The value must be greater than or equal to the 'Output Low' value, specified in the Output menu (section 4.6.6). |
| High Output | Sets a high limit to be imposed whilst autotune is running. The value must be less than or equal to the 'Output High' value, specified in the Output menu (section 4.6.6). |
| State | Read only display of autotune progress:<br>Off. Autotune not running<br>Ready. Fleeting display. Changes immediately to 'Running'.<br>Running. Autotune is in progress.<br>Complete. Autotune completed successfully. This is a fleeting display which changes immediately to 'Off'.<br>Timeout, TI Limit and R2G Limit are error conditions described in Appendix B section B2.4.5. If any of these occurs, tuning is aborted and the PID settings remain unchanged. |
| Stage | A read only display showing the progress of the autotune:<br>Settling. Displayed during the first minute whilst loop stability is checked (Appendix B, section B2.4.5)<br>To SP. Heating or cooling switched on.<br>Wait min. Power output off.<br>Wait max. Power output on.<br>Timeout, TI Limit and R2G Limit are error conditions described in Appendix B section B2.4.5. |
| Stage Time | Time into the current stage of the autotune process. 0 to 99999 seconds. |
| AT.R2G | Autotune at R2G. 'Yes' means that the control loop uses the R2G value calculated by autotune. 'No' causes the loop to use the R2G value entered by the user (PID menu) calculated as described in Appendix B section B2.4.5. |

### 4.6.4 PID menu parameters

Note: If control type is set to 'Off', or 'OnOff' in the Setup menu, the PID menu contains only the Loop Break time parameter 'LBT'.

| | |
|---|---|
| Sched Type | Selects the type of gain scheduling (section B2.3.7) to be applied.<br>Off. Gain scheduling not active<br>Set. The user selects the PID parameter set to be used.<br>Setpoint. Transfer from one set to the next depends on the setpoint value<br>PV. The transfer from one set to another depends on the PV value<br>Error. The transfer between sets depends on the value of the error signal<br>OP. Transfer depends on the value of the output.<br>Rem. Transfer is controlled by a remote input. |
| Number of Sets | Allows the number of sets of PID parameters for use in Gain scheduling to be selected. |
| Remote input | For 'Sched Type' = 'Rem' only, this shows the current value of the remote input channel being used to select which set is active. If the remote input value ≤ the Boundary 1-2 value (see below) then set 1 is selected. If it is > Boundary 1-2 value but ≤ Boundary 2-3 value then set 2 is used. If the remote value is > Boundary 2-3 value, then set three is used. If the Remote input is not 'wired', the value is user editable from the front panel. |
| Active Set | The set number currently in use. |
| Boundary 1-2 | For all Sched Types except 'Set', this allows the user to enter a 'boundary' value, which means that if the relevant value (SP, PV, Error etc.) rises above this boundary, the loop switches from PID set 1 to PID set 2. If it falls below the boundary value, the loop switches from set 2 to set 1. |
| Boundary 2-3 | As above but for switching between sets 2 and 3. |
| PB/PB2/PB3 | Proportional band for set one/two/three. The proportional term in the units (Engineering units or %) set in 'PBUnits' in the Setup menu. See Appendix B section B2.2.2 for more details. |
| Ti/Ti2/Ti3 | Integral time constant for set one/two/three. Valid entries are1 to 9999.9 seconds, or 'Off'. If set Off, then integral action is disabled. Removes steady state control offsets by moving the output up or down at a rate proportional to the error signal. |
| Td/Td2/Td3 | Derivative time constant for set one/two/three. Valid entries are 1 to 9999.9 seconds, or 'Off'. If set Off, then derivative action is disabled. Determines how strongly the controller reacts to a change in the PV. Used to control overshoot and undershoot and to restore the PV rapidly if there is a sudden change in demand. |
| R2G/R2G2/R2G3 | Relative cool gain for set one/two/three. Appears only if cooling has been configured (Ch2 Control not 'Off' or 'OnOff' in Setup menu). Valid entries are 0.1 to 10. Sets the cooling proportional band which compensates for differences between heating and cooling power gains. |
| CBH/CBH2/CBH3 | Cutback high for set one/two/three. Valid entries 'Auto' (3×PB) or 0.1 to 9999.9. The number of display units above setpoint at which the controller output is forced to 0% or -100% (OP min), in order to modify undershoot on cool down. See section B2.3.2 for more details. |
| CBL/CBL2/CBL3 | Cutback low for set one/two/three. Valid entries 'Auto' (3×PB) or 0.1 to 9999.9. The number of display units below setpoint at which the controller output is forced to 100% (OP max), in order to modify overshoot on heat up. See section B2.3.2 for more details. |
| MR/MR2/MR3 | Manual reset for set one/two/three. Valid entries 0 to 100%. Introduces a fixed additional power level to the output in order to eliminate steady state error from proportional only control. Applied instead of the integral component when Ti is set to 'Off'. |
| LBT/LBT2/LBT3 | Loop break time for set one/two/three. valid entries are 1 to 99999 seconds, or 'Off'. See section B2.3.6 for more details. |
| Output Low/2/3 | Output low limit for set one/two/three. Valid entries are in the range Output High/2/3 to -100. |
| Output High/2/3 | Output high limit for set one/two/three. Valid entries are in the range Output Low/2/3 to +100 |

### 4.6.5 Setpoint menu parameters

| | |
|---|---|
| Range High/Low | Range limits. Valid entries from 99999 to -99999. Range limits set absolute maxima and minima for control loop setpoints. If the proportional band is configured as a % span, the span is derived from the range limits. |
| SP select | Select SP1 or SP2. SP1 is considered to be the primary setpoint for the controller, and SP2 a secondary (standby) setpoint. |
| SP1, SP2 | Allows values for Setpoints 1 and 2 to be entered. Valid entries are any within the range 'SPHigh Limit' to 'SPLowLim'. |
| SP Low Limit | Minimum setpoint limit for SP1 and SP2. Valid entries are in the range 'Range Lo' and 'SP High Limit' |
| SP High Limit | Maximum setpoint limit for SP1 and SP2. Valid entries are in the range 'Range Hi' and 'SP Low Limit' |
| Alt SP Enable | 'Yes' enables the alternative setpoint; 'No' disables it. May be wired to an external or internal source. |
| Alt SP | When wired this is a read only display of the alternative setpoint value. Otherwise, the user may insert a value. Valid values are limited by 'Range Hi' and 'Range Lo'. |
| Rate | Sets the maximum rate at which the working setpoint may change in Engineering units per minute. Often used to protect the load from thermal shock cause by large step changes in setpoint. 'Off' disables rate limiting. |
| Rate Done | Read only display. 'Yes' indicates that the working setpoint has completed its change. 'No' indicates that the setpoint is still ramping. |
| SP Rate Disable | Appears only if Rate is not 'Off'. 'Yes' disables rate limiting; 'No' enables rate limiting. |
| Servo To PV | If 'Rate' is set to any value other than 'Off', and if 'Servo to PV' is set to 'Yes' then any change in the current setpoint value causes the working setpoint to servo to the current PV before ramping to the new setpoint value. |
| SP Trim | A positive or negative value added to the setpoint, for local fine tuning. Valid entries are any value between 'SP Trim High' and 'SP Trim Low'. |
| SP Trim High/Low | Setpoint trim high and low limits |
| Manual Track | 'On' enables manual tracking to allow the local SP to follow the value of the current PV. See section B2.5.5 for more details. 'Off' disables manual tracking. |
| Setpoint Track | 'On' enables setpoint tracking to allow the local SP to follow the value of the alternative SP. See section B2.5.4 for more details. 'Off' disables setpoint tracking. |
| Track PV | The unit tracks the PV when it is servoing or tracking. |
| Track Value | The SP to track in manual tracking |
| SP Int Balance | Allows the user to enable (tick) or disable (cross) debump on PV change. |

### 4.6.6 Output menu items

Appendix B section B2.6 contains details of the output functions.

| | |
|---|---|
| Output Low | The minimum power, or the maximum 'negative' (i.e. cooling) power to be delivered by the system. The valid input range is -100% and Output High. |
| Output High | The maximum output power to be delivered by channels 1 and 2, where 100% is full power. The valid input range is Output Low to 100.0%. Reducing this value reduces the rate of change of the process, but it also reduces the controller's ability to react to perturbations. |
| Ch1 Output | Displays the positive power values used by the heat output. Values range from Output low to Output high |
| Ch2 Output | Displays the cooling power values for channel two. Appears as a value between Output high and -100%, where -100% represents full cooling power. |
| Ch2 Deadband | A gap (in %) between output 1 switching off, and output 2 switching on, and *vice-versa*. Valid inputs are 0 (off) to 100%. |
| Rate | Limit on the rate at which the output from the PID can change. Can be useful in preventing rapid changes in output that could damage the process, heater elements etc. |
| Ch1 OnOff Hyst | Appears only if 'Ch1 Control' has been set to 'OnOff' in the Setup menu. Allows the user to enter a hysteresis value for channel one. Valid entries are 0.0 to 200.0. |
| Ch2 OnOff Hyst | Appears only if 'Ch2 Control' has been set to 'OnOff' in the Setup menu. Allows the user to enter a hysteresis value for channel two. Valid entries are 0.0 to 200.0. |
| Ch1 Travel Time | Appears only if Setup menu parameter 'Ch1 Control' is set to 'VPB' or 'VPU'. This is the valve travel time from closed (0%) to open (100%). In a valve positioning application, channel 1 output is connected by a single software 'wire' to a Valve Raise/Valve Lower relay pair. For heat/cool applications, channel 1 is associated with the heating valve. Valid entries: 0.0 to 1000.0 seconds. |
| Ch2 Travel Time | Appears only if Setup menu parameter 'Ch2 Control' is set to 'VPB' or 'VPU'. This is the valve travel time from closed (0%) to open (100%). For heat/cool applications, channel 2 is associated with the cooling valve. Valid entries: 0.0 to 1000.0 seconds. |
| Nudge Raise | Appears only if Setup menu parameter 'Ch1 Control' or Ch2 Control is set to 'VPU'. If set to 'Yes', the valve can be moved towards the open position by, for example, a contact closure, an up arrow button operation or a serial communications command. The default minimum nudge time is 125 ms, but this can be edited in the relevant relay configuration - see section 4.11.2. See also Section B2.6.10 for more 'Nudge' details. |
| Nudge Lower | As for 'Nudge Raise', above but moves the valve towards the closed position. |
| Ch1 Pot Pos* | The position of the channel one actuator as measured by the feedback potentiometer. |
| Ch1 Pot Brk* | 'On' indicates that the input to the relevant channel is open circuit. |
| Ch2 Pot Pos* | The position of the channel two actuator as measured by the feedback potentiometer. |
| Ch2 Pot Brk* | 'On' indicates that the input to the relevant channel is open circuit. |
| Pot Brk Mode* | Defines the action to be taken if a potentiometer break is detected:<br>Raise: opens the valve<br>Lower: closes the valve<br>Rest: the valve remains in its current state.<br>Model: the controller tracks the position of the valve and sets up a model of the system so that it continues to function if the potentiometer becomes faulty. |

* Note: These parameters appear only if the 'Setup' menu parameter 'Ch1 Control' or 'Ch2 control' (as appropriate) is set to 'VBP'. The Setup menu is described in section 4.6.2.

### 4.6.6 OUTPUT MENU PARAMETERS (Cont.)

| | |
|---|---|
| SBrk Mode | Defines the action to be taken in the event of a sensor break.<br>Safe: The output adopts the value configured in 'Sbrk OP', below.<br>Hold: The output remains at its current level. |
| Sbrk OP | The value to be output if a sensor break occurs, and SBrk Mode (above) is set to 'Safe'. |
| Safe OP | The output level adopted when the loop is inhibited (Main menu section 4.6.1). |
| Manual Mode | Selects the type of transition to occur when changing to manual mode (section 4.6.1):<br>Track: Whilst in Auto mode, the manual output tracks the control output so that there is no change of output when manual mode is switched to.<br>Step: On transition to manual mode, the output is set to the value entered for 'Forced-OP' (below).<br>Last Man. Out: On transition to manual mode, the output adopts the manual output value as last set by the operator. |
| Manual Output | The output when the loop is in manual mode. In manual mode the controller limits the maximum power, but it is not recommended that it be left unattended at high power settings. It is important that over range alarms are fitted to protect the process. |

Note: It is recommended that all processes are fitted with an independent over range detection system.

| | |
|---|---|
| Forced Output | Forced Manual output value. When 'Manual Mode' = 'Step', this is the output value adopted when changing from Auto to Manual mode. |
| Manual Startup | When set to off (cross symbol), the controller powers up in the same (auto or manual) mode that obtained when it was switched off. When set to on (tick symbol) the controller always powers up in manual mode. |
| Pff En | Power feed forward enable. 'Yes' enables power feed forward (adjusts the output signal to compensate for variations is supply voltage. 'No' disables Pff. See section B2.6.6 for further details. |
| Power In | Read only display of the current supply voltage. |
| Cool Type | Appears only if 'Ch2 Control' = 'PID' in the setup menu (section 4.6.2) and allows the user to enter the appropriate type of cooling (section B2.6.7):<br>Linear: For use when controller output changes linearly with PID demand.<br>Oil: For oil cooled applications<br>Water: For water cooled applications<br>Fan: For forced air cooling. |
| FF Type | Feed forward type (section B2.6.8):<br>None: No signal fed forward.<br>Remote: A remote signal fed forward.<br>SP: Setpoint is fed forward.<br>PV: PV is fed forward. |
| FF Gain | For FF types 'PV' and 'SP', this scales the feed forward signal. |
| FF Offset | For FF types 'PV' and 'SP', this defines the offset of the scaled feed forward signal. |
| FF Trim lim | For FF types 'PV' and 'SP', defines symmetrical limits about the PID output which are applied to the scaled feed forward signal. |
| FF OP | For FF types 'PV' and 'SP', this is the calculated (scaled, offset and trimmed) feed forward signal. FF OP = FF gain (input + FF Offset) |
| Track Output | If 'Track Enable' (below) is set to 'Yes', this is the value for the control output. PID remains in Auto mode and tracks the output. The Track OP value can be wired to an external source, or can be entered via the front panel. Similar to entering manual mode. |
| Track Enable | When set to 'Yes', the output follows the Track OP value (above). When subsequently set to 'Off' the loop makes a bump less return to control. |
| Rem. Output Low/High | Used to limit the output using a remote source. These limits cannot exceed the 'Output Low' and 'Output High' values described earlier in this section. |

### 4.6.7 Loop diagnostics

These 'parameters' are read only unless otherwise stated.

| | |
|---|---|
| Error | The difference in value between the setpoint and the PV. |
| Target Output | The requested control output. The target of the active output if rate limiting is active. |
| Working Out Low | The low limit for the working output. This is the value used to limit the output power of the loop and is derived from the gain scheduled limit, the remote limit and the safety limit. |
| Working Out High | The high limit for the working output. This is the value used to limit the output power of the loop and is derived from the gain scheduled limit, the remote limit and the safety limit. |
| Loop Break | Loop Break Alarm. Becomes active 'Yes' if the loop break time (LBT), set in the PID menu (section 4.6.4) is exceeded, otherwise 'No' is displayed. |
| Prop. Output | Shows the proportional term contribution to the control output |
| Integral Output | Shows the integral term contribution to the control output |
| Deriv. Output | Shows the derivative term contribution to the control output |
| Sensor Break | Indicates sensor break status. On (tick symbol) indicates a sensor break has occurred; Off (cross symbol) shows that no sensor breaks have been detected. |
| Sched PB | The scheduled proportional band for the current PID set. |
| Sched Ti | The scheduled integral time for the current PID set. |
| Sched Td | The scheduled derivative time for the current PID set. |
| Sched R2G | The scheduled relative cool gain value for the current PID set. |
| Sched CBH | The scheduled cutback high value for the current PID set. |
| Sched CBL | The scheduled cutback low value for the current PID set. |
| Sched MR | The scheduled manual reset value for the current PID set. |
| Sched Loop Break | The scheduled loop break time for the current PID set. |
| Sched Out Low | The scheduled output low limit for the current PID set. |
| Sched Out High | The scheduled output high limit for the current PID set. |

## 4.7 ADVANCED LOOP CONFIGURATION

Similar to the Loop option described above, advanced loop includes the ability to run a cascade loop. Figure 4.7 is an overview of the configuration menu structure.

Advanced Loop

- Main (Section 4.7.1)
  - Master PV
  - Master WSP
  - Target setpoint
  - Slave PV
  - Slave WSP
  - Working Output
  - Cascade Mode
  - Inhibit
  - Master Int.Hold
  - Slave Int.Hold
- Setup (Section 4.7.2)
  - Master Name
  - Slave Name
  - Master Loop
  - Cascade Type
  - Slave Channel 1
  - Slave Channel 2
  - Setpoint access
  - Mode Access
- Tune (Section 4.7.3)
  - Tune Type
  - Tune Enable
  - Tune Slave R2G
  - Tune High
  - Tune Low
  - Output High
  - Output Low
  - State
  - Stage Time
  - Diagnostics
- Master PID (Section 4.7.4)
  - Control Action
  - PB Units
  - Derivative Type
  - PB
  - Ti
  - Td
  - CBH
  - CBL
  - MR
  - LBT
- Slave PID (Section 4.7.5)
  - Control Action
  - PB Units
  - Derivative Type
  - Sched Type
  - No of sets
  - Remote input
  - Set boundaries
  - PB
  - Ti
  - Td
  - R2G
  - CBH
  - CBL
  - MR
  - LBT
  - Output Low
  - Output High
- Master SP (Section 4.7.6)
  - Range High
  - Range Low
  - SP Select
  - SP1
  - SP2
  - SP High Limit
  - SP Low Limit
  - Alt SP Enable
  - Alt SP
  - Rate
  - Rate Done
  - SP Trim
  - SP Trim High
  - SP Trim Low
  - Manual Track
  - Setpoint Track
  - Track PV
  - Track SP
  - SP Int Balance
- Slave SP (Section 4.7.7)
  - Range High
  - Range Low
  - SP High Limit
  - SP Low Limit
  - Local SP
  - Alt SP
  - Remote FF
  - Remote FF Enable
  - Remote FF High
  - Remote FF Low
  - Manual Track
- Output (Section 4.7.8)
  - Output High
  - Output Low
  - Ch1 Output
  - Ch2 Output
  - Ch2 Deadband
  - Rate
  - Sbrk Mode
  - Sbrk OP
  - Safe OP
  - Manual Mode
  - Manual Output
  - Forced Output
  - Manual Startup
  - Pff En
  - Line Voltage
  - Cool Type
  - FF Type
  - Track Output
  - Track Enable
  - Rem. Output Low
  - Rem. Output High
- Diagnostics (Section 4.7.9)
  - Master Eror
  - Slave Error
  - (M)Prop. Output
  - (M)Integral Out
  - (M)Deriv. Output
  - (S)Prop. Output
  - (S)Integral Out
  - (S)Deriv. Output
  - Target Output
  - Loop Break (S)
  - Loop Break (M)
  - Sensor Break (S)
  - Sensor Break (M)
  - Sched PB
  - Sched Ti/Td
  - Sched CBH/CBL
  - Sched MR
  - Sched Loop Break
  - Sched R2G
  - Sched Out High
  - Sched Out Low
  - Working Out Low
  - Working Out High

### 4.7.1 Advanced Loop Main menu

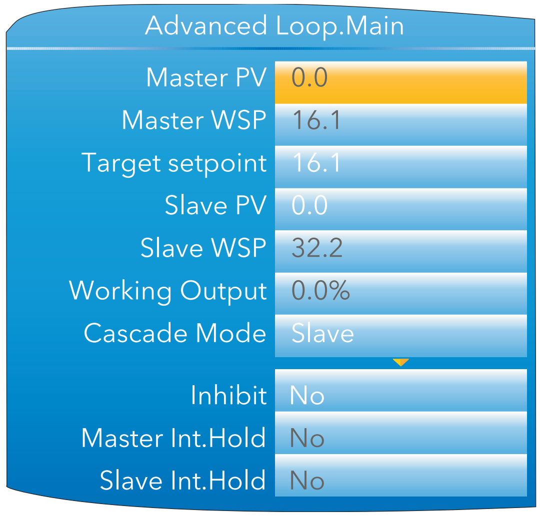



Figure 4.7.1 Main menu

| | |
|---|---|
| Master PV | This is the process value for the outer (master) loop of cascade control, typically obtained from an analogue input. |
| Master WSP | This is the (read only) working setpoint for the outer (master) loop of cascade control. The Master WSP can obtain its value from one of a number of sources such as 'Internal SP' or 'Remote SP'. |
| Target setpoint | The target setpoint is the value which the outer (master) control loop is attempting to reach. The value may come from one of a number of sources, such as internal SP or remote SP |
| Slave PV | This is the process value for the inner (slave) loop of cascade control, typically wired from an analogue input. |
| Slave WSP | This is the (read only) working setpoint for the inner (slave) loop. The value may come from one of a number of sources, such as the output from the master loop or the local slave setpoint. |
| Working Output | The actual output of the inner (slave) loop before it is split into channel 1 and channel 2 outputs. |
| Cascade Mode | Slave: Also known as 'Slave Local Auto', this is a single loop controlling with a local setpoint.<br>Manual: Also known as 'Slave Manual', this provides a single manual power setting for the slave.<br>Cascade: (Full) cascade. In this mode, the master is in 'Auto' mode and provides the setpoint for the slave. |
| Inhibit | If set to 'Yes', both outer (master) loop and inner (slave) loops stop controlling and the output of the slave loop is set to the safe output value (SafeOp) set in the Output menu (section 4.7.8). |
| Master Int.Hold | If set to 'Yes', the integral component of the outer (master) loop PID calculation is held at its current value and does not integrate any further disturbances in the plant. Essentially this is equivalent to switching into PD control with a manual reset value pre-configured. |
| Slave Int.Hold | As for Master.IntHold, above, but for the inner (slave) loop. |

### 4.7.2 Advanced Loop Setup menu

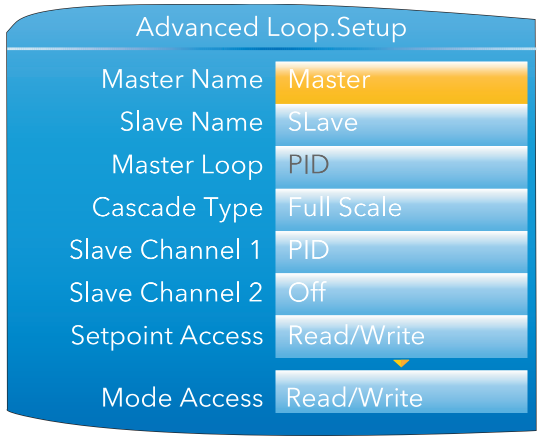



Figure 4.7.2 Advanced Loop Setup menu

| | |
|---|---|
| Master Name | Allows the user to enter a 10-character string for the Master loop name in the Cascade display page (section 3.4.8) |
| Slave Name | As above, but for the slave loop. |
| Master Loop | The control algorithm for the master control loop (PID only for this software release). |
| Cascade Type | Full Scale: The master generates a setpoint (between SP High limit and SP Low limit) for the slave.<br>Trim: The master working setpoint is used as the base setpoint of the slave. This is then modified by the addition of a setpoint trim, to become the target setpoint for the slave. The PID output from the master is mapped to range set by Trim Range High and Trim Range Low. |
| Slave Channel 1 | Selects the channel 1 control algorithm. Different algorithms can be selected for channels 1 and 2. In temperature control applications, channel 1 is usually the heating channel, and channel 2 the cooling channel.<br>PID: Control Output Configured as PID<br>VPB: Control Output Configured as Bounded VP. Bounded VP is implemented as a PID algorithm driving a position loop and is used in systems with position feedback. |
| Slave Channel 2 | Selects the channel 2 control algorithm. Different algorithms can be selected for channels 1 and 2. In temperature control applications, channel 1 is usually the heating channel, channel 2 the cooling channel.<br>Off: Control output is not configured<br>PID: Control Output Configured as PID |
| Setpoint Access | Allows the user to select 'Read Only', 'Read/Write', or 'Operator R/W' for setpoint access, where 'Operator R/W means that the setpoint is read write for access levels operator and above, but read only in Logged out mode. |
| Mode Access | As for 'Setpoint Access', above, but for Auto/Manual mode switching. |

### 4.7.3 Advanced Loop Tune menu

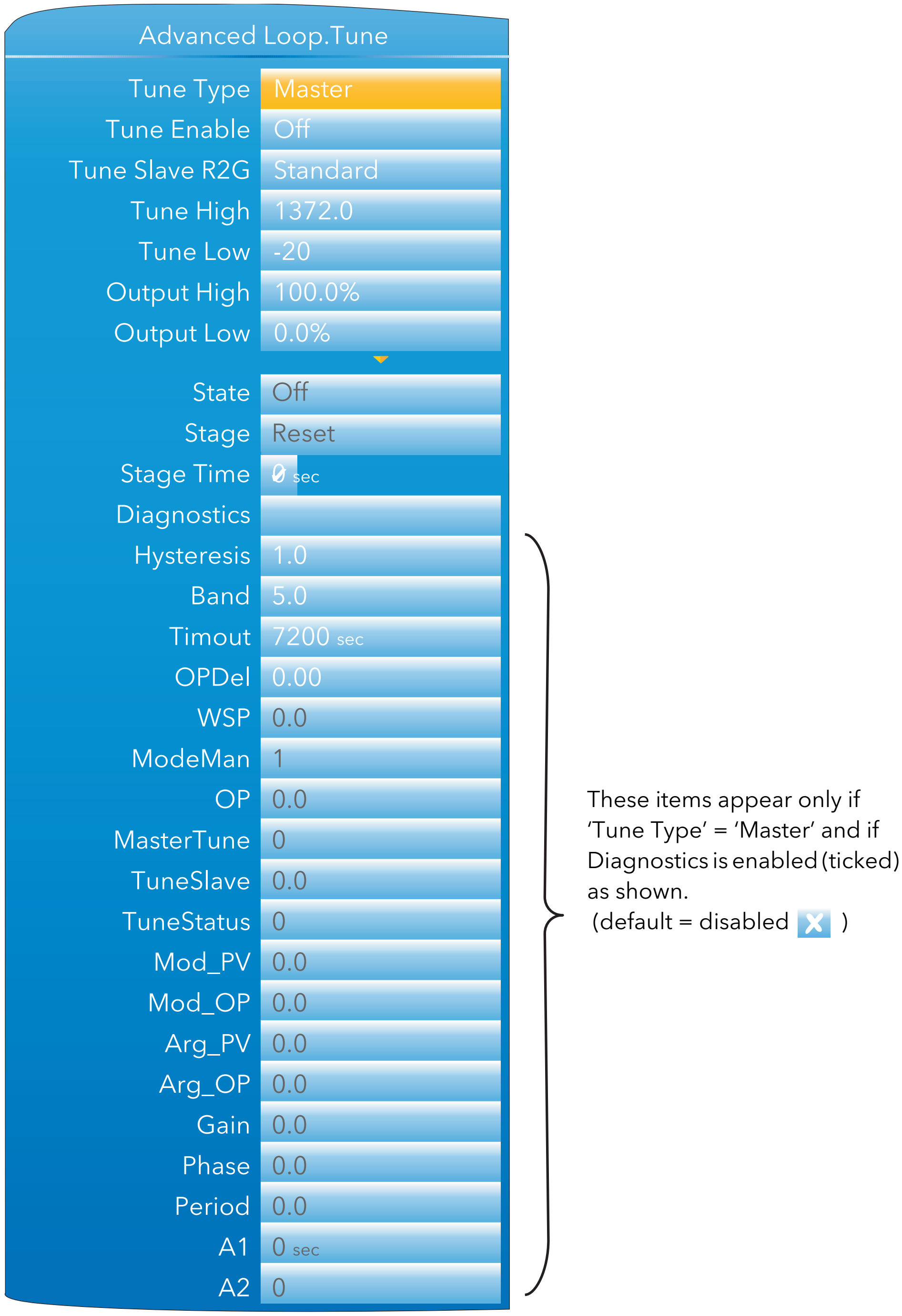



Figure 4.7.3 Advanced Loop Tune menu

| | |
|---|---|
| Tune Type | Select 'Master' or 'Slave' for the Tuning process. |
| Tune Slave R2G | Appears only if the Slave channel 2 is set to 'PID' in the Setup menu (section 4.7.2), and Tune Type is set to Slave in the Advanced Loop.Tune menu.<br>Standard: Normal compensation applied to account for differences in heating and cooling efficiencies between the heating and cooling channels.<br>R2GPD: Typically used in heavily lagged systems. |
| Tune Enable | Allows the user to initiate an autotune. |

### 4.7.3 ADVANCED LOOP TUNE MENU (Cont.)

| | |
|---|---|
| Tune High | Sets the maximum value for the master loop setpoint during the tuning process. |
| Tune Low | Sets the maximum value for the master loop setpoint during the tuning process. |
| Output High | The maximum output power level which the controller may supply during the tuning process. If 'Output High' in the Output menu (section 4.7.8) is lower than 'High Output' then the maximum output is clipped to the 'Output High' value. |
| Output Low | The minimum output power level which the controller may supply during the tuning process. If 'Output Low' in the Output menu ((section 4.7.8) is higher than 'Low Output' then the minimum output is clipped to the 'Output Low' value. |
| State | The current autotune state.<br>Off: Autotune not enabled<br>Ready:<br>Running: Autotune running<br>Complete: The tune process completed successfully. Fleeting display before returning to 'Off'.<br>Time-Out: A timeout error has occurred and the autotune has been aborted.<br>Ti Limit<br>R2G Limit |
| Stage | Reset<br>None<br>Settling<br>Current SP<br>New SP<br>To SP<br>Wait Max<br>Wait Min<br>Store<br>CoolT<br>PID<br>Abort<br>Complete<br>NewR2G<br>1:Half Cycle<br>2:Full Cycle<br>3:Full Cycle<br>4:FinalCycle<br>5:Calc. |
| Stage Time | Elapsed time since entering this stage of the tuning. |
| Diagnostics | If this is enabled, a number of further parameters become visible. |
| Hysteresis | This defines the hysteresis of the switch used during master autotuning to generate the oscillation. It is set as a % of the master PV range (High Range - Low Range) in engineering units being +/- Hysteresis/2 about the tuning setpoint |

### 4.7.3 ADVANCED LOOP TUNE MENU (Cont.)

| | |
|---|---|
| Band | This defines the band between which the setpoint of the slave controller will be switched during the master autotune oscillation. It is set as a % of the master PV (High Range - Low Range) in engineering units being +/- Band/2 about the tuning setpoint. The actual values applied to the slave may actually be constrained inside this band by the wind-up control mechanism |
| Timeout | Defines the maximum time permitted for each stage of the master tuning. |
| OPDel | This is an internal setting of the order of 0.5 during tuning. |
| WSP | This is the actual setpoint around which the autotuning oscillation of the master takes place. It is used for the calculations associated with the Hysteresis and Band parameters. |
| ModeMan | This parameter is used by the master autotune algorithm to communicate with the master loop. Puts master controller into 'Not-Auto' mode |
| OP | This signal is generated within the master loop during the autotune oscillation. It is used only as an input to the calculations which generate the slave loop setpoint. It is not the overall loop output to the load which at all times is under the control of the slave PID calculations. |
| MasterTune | Master tune in progress |
| TuneSlave | The autotune process is requesting a slave tune. |
| Tune Status | This indicates the internal stage of tuning.<br>0 = Not tuning 1 = Tuning the slave<br>2 = Tuning the master 3 = Tuning completed<br>-1 = Tuning has aborted or timed-out |
| Mod_PV | This is the amplitude of the fundamental component of the master PV during the last cycle of the tuning oscillation. |
| Mod_OP | This is the amplitude of the fundamental component of the master OP during the last cycle of the tuning oscillation. |
| Arg_PV | This is the argument (phase) of the fundamental component of the master PV during the last cycle of the tuning oscillation. Value in radians. |
| Arg_OP | This is the argument (phase) of the fundamental component of the master OP during the last cycle of the tuning oscillation. Value in radians. |
| Gain | This is the gain between the master OP and the master PV over the path via the slave loop and the load, measured at the fundamental frequency of the autotuning oscillation. |
| Phase | The phase shift in radians between the master OP and the master PV over the path via the slave loop and the load, measured at the fundamental frequency of the autotuning oscillation |
| Period | This is the period of the last cycle of the autotune oscillation, in seconds. |
| A1 | This is the number of samples actually taken in order to determine the fundamental components of the master PV and OP. The target number is around 100 samples but the actual number taken may differ slightly from this depending on the load's behaviour. |
| A2 | The A2 parameter is a used for diagnostic purposes. Its value indicates the design method chosen by the algorithm which depends on the characteristics of the master tuning oscillation and the measured values of frequency, gain, and phase shift around the master loop. This influences the choices of the P, I and D values set into the master loop. |
| Alpha_p | R2GPD tuning diagnostic parameter: Heat time / cool time. |
| OPss | R2GPD tuning diagnostic parameter: Steady state output at the end of the settling period. |
| Alpha | R2GPD tuning diagnostic parameter: 1/R2G. |
| Debug | R2GPD tuning diagnostic parameter: 0-PID, 1-PI, 2-PD, 3-P. |
| CycleNo | R2GPD tuning diagnostic parameter: Number of cycles in auto tune sequence. |

### 4.7.3 ADVANCED LOOP TUNE MENU (Cont.)

| | |
|---|---|
| CycleNo | R2GPD tuning diagnostic parameter: Number of cycles in auto tune sequence. |
| PBs | R2GPD tuning diagnostic parameter: PBs scales the proportional band which will be used in the PD settling period. |
| TDs | R2GPD tuning diagnostic parameter: TDs scales the derivative value which will be used during the PD settling period. |
| Settle | R2GPD tuning diagnostic parameter: Used to scale the last cycle time. The result will be used for the PD settling time. |

### 4.7.4 Advanced Loop Master PID menu

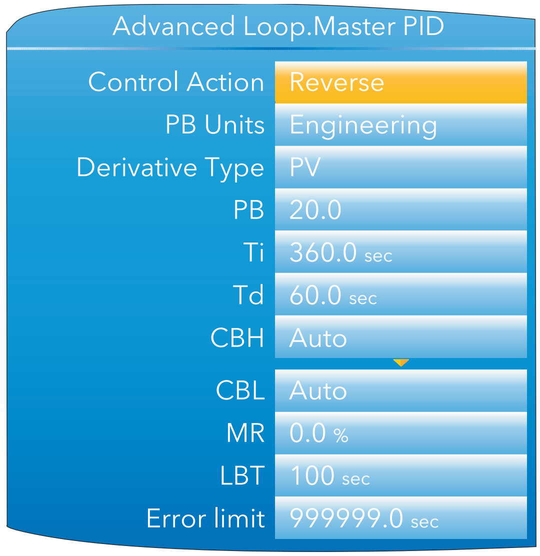



Figure 4.7.4 Advanced Loop master PID menu

| | |
|---|---|
| Control Action | Select 'Reverse' or 'Direct'.<br>'Reverse' means that the output is 'on' when the process value (PV) is below the target setpoint (SP). This is normal for heating control.<br>'Direct' means that the output is on when PV is above SP. This is normal for cooling control. |
| PB Units | Select 'Engineering' or 'Percent'.<br>'Engineering' displays values in (for example) temperature units (e.g. °C or °F).<br>'Percent' displays values as a percentage of loop span (Range High - Range Low). |
| Deriv Type | 'Error' means that changes to PV or SP cause changes to the derivative output. Derivative on error should be used with a programmer since it tends to reduce ramp overshoot. 'Error' provides rapid response to small setpoint changes which makes it ideal for temperature control systems.<br>'PV' means that changes in PV alone cause changes to the derivative output. Typically used for process systems using valve control, as it reduces wear on the valve mechanics. |
| PB | Proportional band. The proportional term in the units (Engineering units or %) set in 'PBUnits' above. See Appendix B section B2.2.2 for more details. |
| Ti | Integral time constant. Valid entries are1 to 9999.9 seconds, or 'Off'. If set Off, then integral action is disabled. Removes steady state control offsets by moving the output up or down at a rate proportional to the error signal. |
| Td | Derivative time constant. Valid entries are 1 to 9999.9 seconds, or 'Off'. If set Off, then derivative action is disabled. Determines how strongly the controller reacts to the rate-of-change in the PV. Used to control overshoot and undershoot and to restore the PV rapidly if there is a sudden change in demand. |
| CBH | Cutback high. Valid entries 'Auto' (3×PB) or 0.1 to 9999.9. The number of display units above setpoint at which the controller output is forced to 0% or -100% (OP min), in order to modify undershoot on cool down. See section B2.3.2 for more details. |
| CBL | Cutback low. Valid entries 'Auto' (3×PB) or 0.1 to 9999.9. The number of display units below setpoint at which the controller output is forced to 100% (OP max), in order to modify overshoot on heat up. See section B2.3.2 for more details. |
| MR | Manual reset. Valid entries -100% to +100%. Introduces a fixed additional power level to the output in order to eliminate steady state error from proportional only control. Applied instead of the integral component when Ti is set to 'Off'. |
| LBT | Loop break time. valid entries are 1 to 99999 seconds, or 'Off'. See section B2.3.6 for more details |

### 4.7.5 Advanced Loop Slave PID menu

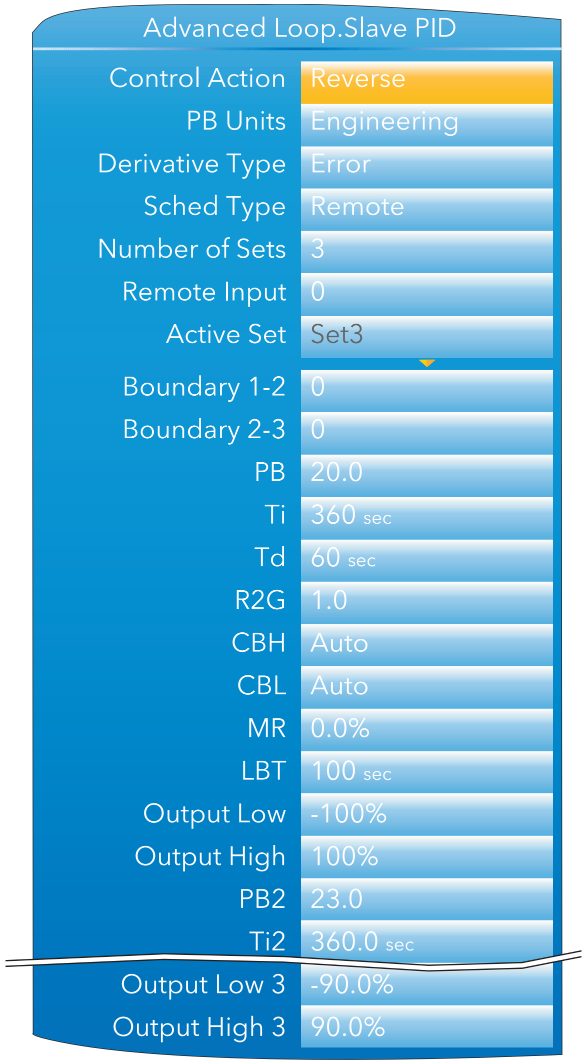



Figure 4.7.5 Advanced Loop Slave PID Menu (Typical)

Control Action
: Select 'Reverse' or 'Direct'.
'Reverse' means that the output is 'on' when the process value (PV) is below the target setpoint (SP). This is normal for heating control.
'Direct' means that the output is on when PV is above SP. This is normal for cooling control.

PB Units
: Select 'Engineering' or 'Percent'.
'Engineering' displays values in (for example) temperature units (e.g. °C or °F).
'Percent' displays values as a percentage of loop span (Range High - Range Low).

Deriv Type
: 'Error' means that changes to PV or SP cause changes to the derivative output. Derivative on error should be used with a programmer since it tends to reduce ramp overshoot. 'Error' provides rapid response to small setpoint changes which makes it ideal for temperature control systems.
'PV' means that changes in PV alone cause changes to the derivative output. Typically used for process systems using valve control, as it reduces wear on the valve mechanics.

### 4.7.5 ADVANCED LOOP SLAVE PID MENU (Cont.)

| | |
|---|---|
| Sched Type | Selects the type of Gain Scheduling (section B2.3.7) to be applied.<br>Off. Gain Scheduling not active<br>Set. The user selects the PID parameter set to be used.<br>Setpoint. Transfer from one set to the next depends on the setpoint value<br>PV. The transfer from one set to another depends on the PV value<br>Error. The transfer between sets depends on the value of the error signal<br>OP. Transfer depends on the value of the output.<br>Rem. Transfer is controlled by a remote input. |
| Number of Sets | Allows the number of sets of PID parameters for use in Gain scheduling to be selected. |
| Remote input | For 'Sched Type' = 'Rem' only, this shows the current value of the remote input channel being used to select which set is active. If the remote input value ≤ the Boundary 1-2 value (see below) then set 1 is selected. If it is > Boundary 1-2 value but ≤ Boundary 2-3 value then set 2 is used. If the remote value is > Boundary 2-3 value, then set three is used. If the Remote input is not 'wired', the value is user editable from the front panel. |
| Active Set | The set number currently in use. |
| Boundary 1-2 | For all Sched Types except 'Set', this allows the user to enter a 'boundary' value, which means that if the relevant value (SP, PV, Error etc.) rises above this boundary, the loop switches from PID set 1 to PID set 2. If it falls below the boundary value, the loop switches from set 2 to set 1. |
| Boundary 2-3 | As above but for switching between sets 2 and 3. |
| PB/PB2/PB3 | Proportional band for set one/two/three. The proportional term in the units (Engineering units or %) set in 'PBUnits' in the Setup menu. See Appendix B section B2.2.2 for more details. |
| Ti/Ti2/Ti3 | Integral time constant for set one/two/three. Valid entries are1 to 9999.9 seconds, or 'Off'. If set Off, then integral action is disabled. Removes steady state control offsets by moving the output up or down at a rate proportional to the error signal. |
| Td/Td2/Td3 | Derivative time constant for set one/two/three. Valid entries are 1 to 9999.9 seconds, or 'Off'. If set Off, then derivative action is disabled. Determines how strongly the controller reacts to the rate-of-change in the PV. Used to control overshoot and undershoot and to restore the PV rapidly if there is a sudden change in demand. |
| R2G/R2G2/R2G3 | Relative cool gain for set one/two/three. Appears only if cooling has been configured (Ch2 Control not 'Off' in the Setup menu). Valid entries are 0.1 to 10. Sets the cooling proportional band which compensates for differences between heating and cooling power gains. |
| CBH/CBH2/CBH3 | Cutback high for set one/two/three. Valid entries 'Auto' (3×PB) or 0.1 to 9999.9. The number of display units above setpoint at which the controller output is forced to 0% or -100% (OP min), in order to modify undershoot on cool down. See section B2.3.2 for more details. |
| CBL/CBL2/CBL3 | Cutback low for set one/two/three. Valid entries 'Auto' (3×PB) or 0.1 to 9999.9. The number of display units below setpoint at which the controller output is forced to 100% (OP max), in order to modify overshoot on heat up. See section B2.3.2 for more details. |
| MR/MR2/MR3 | Manual reset for set one/two/three. Valid entries 0 to 100%. Introduces a fixed additional power level to the output in order to eliminate steady state error from proportional only control. Applied instead of the integral component when Ti is set to 'Off'. |
| LBT/LBT2/LBT3 | Loop break time for set one/two/three. valid entries are 1 to 99999 seconds, or 'Off'. See section B2.3.6 for more details. |
| Output Low/2/3 | Output low limit for set one/two/three. Valid entries are in the range Output High/2/3 to -100. |
| Output High/2/3 | Output high limit for set one/two/three. Valid entries are in the range Output Low/2/3 to +100 |

### 4.7.6 Advanced Loop Master SP menu

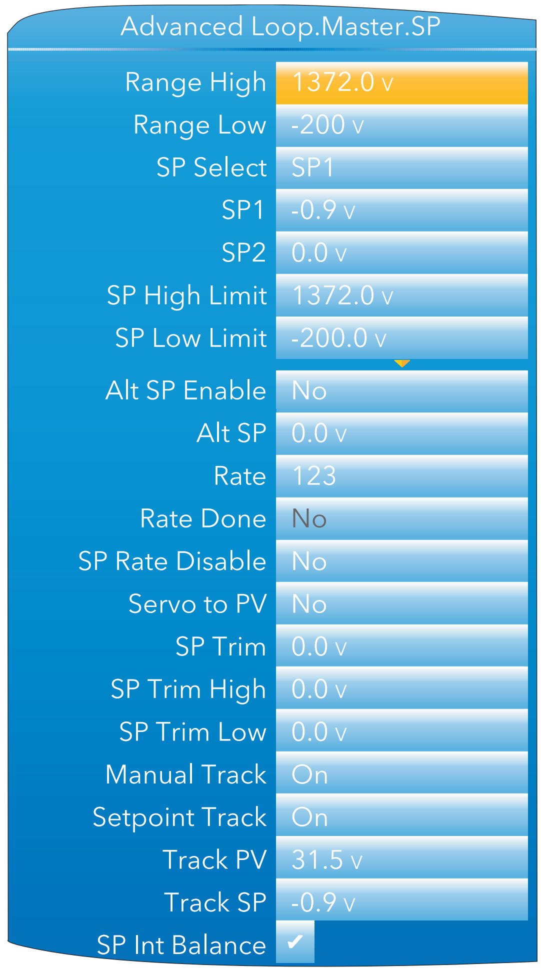



Figure 4.7.6 Advanced Loop Master SP menu

| | |
|---|---|
| Range High/Low | Range limits. Range limits set absolute maxima and minima for control loop setpoints. If the proportional band is configured as a % span, the span is derived from the range limits. |
| SP select | Select SP1 or SP2. SP1 is often considered to be the primary setpoint for the controller, and SP2 a secondary setpoint. |
| SP1, SP2 | Allows values for Setpoints 1 and 2 to be entered. Valid entries are any within the range 'SPHigh Limit' to 'SPLowLim'. |
| SP High Limit | Maximum setpoint limit for SP1 and SP2. Valid entries are in the range 'Range Hi' and 'SP Low Limit' |
| SP Low Limit | Minimum setpoint limit for SP1 and SP2. Valid entries are in the range 'Range Lo' and 'SP High Limit' |
| Alt SP Enable | 'Yes' enables the alternative setpoint; 'No' disables it. May be wired to an external or internal source. |
| Alt SP | When wired this is a read only display of the alternative setpoint value. Otherwise, the user may insert a value. Valid values are limited by 'Range Hi' and 'Range Lo'. |
| Rate | Sets the maximum rate at which the working setpoint may change in Engineering units per minute. Often used to protect the load from thermal shock caused by large step changes in setpoint. 'Off' disables rate limiting. |
| Rate Done | Read only display. 'Yes' indicates that the working setpoint has completed its change. 'No' indicates that the setpoint is still ramping. |

### 4.7.6 ADVANCED LOOP MASTER SP MENU (Cont.)

| | |
|---|---|
| SP Rate Disable | Appears only if Rate is not 'Off'. 'Yes' disables rate limiting; 'No' enables rate limiting. |
| Servo To PV | If 'Rate' is set to any value other than 'Off', and if 'Servo to PV' is set to 'Yes' then any change in the current setpoint value causes the working setpoint to servo to the current PV before ramping to the new setpoint value. |
| SP Trim | A positive or negative value added to the setpoint, for local fine tuning. Valid entries are any value between 'SP Trim High' and 'SP Trim Low'. |
| SP Trim High/Low | Setpoint trim high and low limits |
| Manual Track | 'On' enables manual tracking. Manual tracking removes steps in setpoint when switching between M'Man' and 'Auto' modes. When the loop is switched from manual to auto the target setpoint is set to the current PV. See section B2.5.5 for more details. 'Off' disables manual tracking. |
| Setpoint Track | 'On' enables setpoint tracking. When setpoint tracking is enabled, it ensures 'bumpless' transfer in setpoint when seitching from Alternative setpoint to a local setpoint. See section B2.5.4 for more details. 'Off' disables setpoint tracking. |
| Track PV | The unit tracks the PV when it is servoing or tracking. |
| Track SP | The SP to track in manual tracking - see 'Setpoint Track', above. |
| SP Int Balance | Allows the user to enable (tick) or disable (cross) debump on PV change. |

### 4.7.7 Advanced Loop Slave SP menu

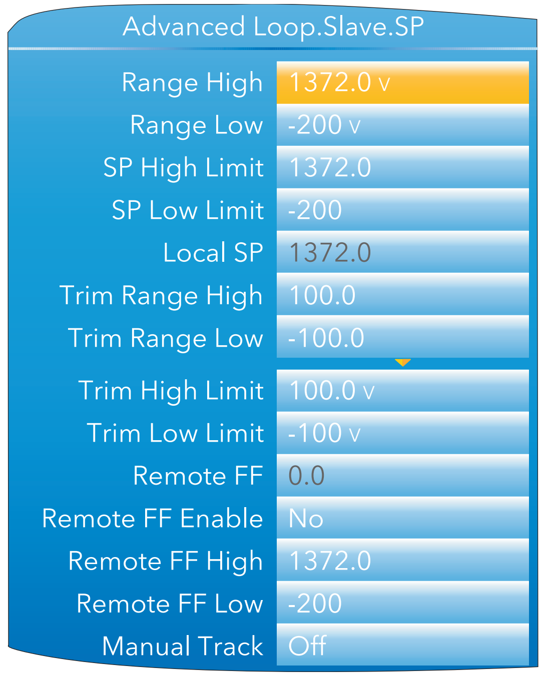



Figure 4.7.7a Advanced Loop Slave Setpint menu

| | |
|---|---|
| Range High/Low | Range limits. Valid entries from 99999 to -99999. Range limits set absolute maxima and minima for control loop setpoints. If the proportional band is configured as a % span, the span is derived from the range limits. |
| SP High Limit | Maximum setpoint limit for the local setpoint. Valid entries are in the range 'Range Hi' and 'SP Low Limit' |
| SP Low Limit | Minimum setpoint limit for the local setpoint. Valid entries are in the range 'Range Lo' and 'SP High Limit' |
| Local SP | The Slave local setpoint |
| Trim Range High | Trim Range upper limit. Appears only if 'Cascade type' has been set to 'Trim' in the Set-up menu. |
| Trim Range Low | Trim Range upper limit. Appears only if 'Cascade type' has been set to 'Trim' in the Set-up menu. |
| Trim High Limit | Maximum value for Trim High value. Appears only if 'Cascade type' has been set to 'Trim' in the Setup menu. |
| Trim Low Limit | Minimum value for Trim Low value. Appears only if 'Cascade type' has been set to 'Trim' in the Setup menu. |
| Remote FF | The current remote feedforward value |
| Remote FF Enable | Enables or disables the use of a remote Feedforward signal. Appears only if 'Cascade type' has been set to 'Full Scale' in the Setup menu. |
| Remote FF High | High limit for the remote feedforward signal value. Appears only if 'Cascade type' has been set to 'Full Scale' in the Setup menu. |
| Remote FF Low | Low limit for the remote feedforward signal value. Appears only if 'Cascade type' has been set to 'Full Scale' in the Setup menu. |
| FF Select | Allows the user to select the source of the feedforward signal from 'master PV', Master working setpoint' or Remote FF'. Appears only if 'Cascade type' has been set to 'Trim' in the Setup menu. |
| Manual Track | 'On' enables manual tracking to allow the local SP to follow the value of the current PV to allow bumpless transfer when switching to Auto. See section B2.5.5 for more details. 'Off' disables manual tracking. |

### 4.7.7 ADVANCED LOOP SLAVE SP MENU (Cont.)

| | |
|---|---|
| Sbrk Mode | This defines the behaviour when the master loop process variable is bad, i.e. the sensor has failed. |
| Sbrk SP | The setpoint for the slave loop when the master sensor has gone into sensor break and the sensor break mode for the master is set to SbrkSP |

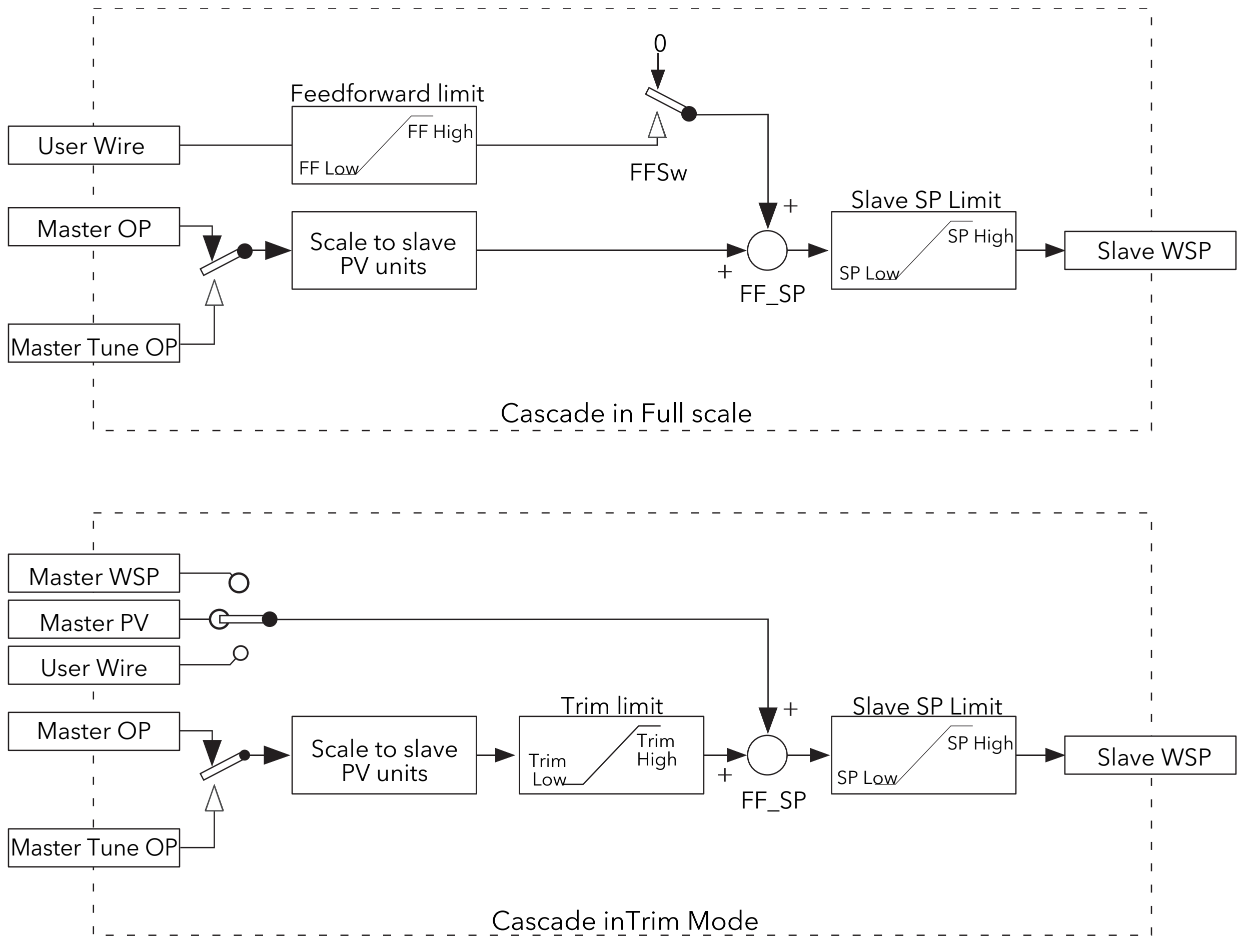



Figure 4.7.7b Slave Working setpoint limits

### 4.7.8 Advanced Loop Output menu

Appendix B section B2.6 contains details of the output functions.

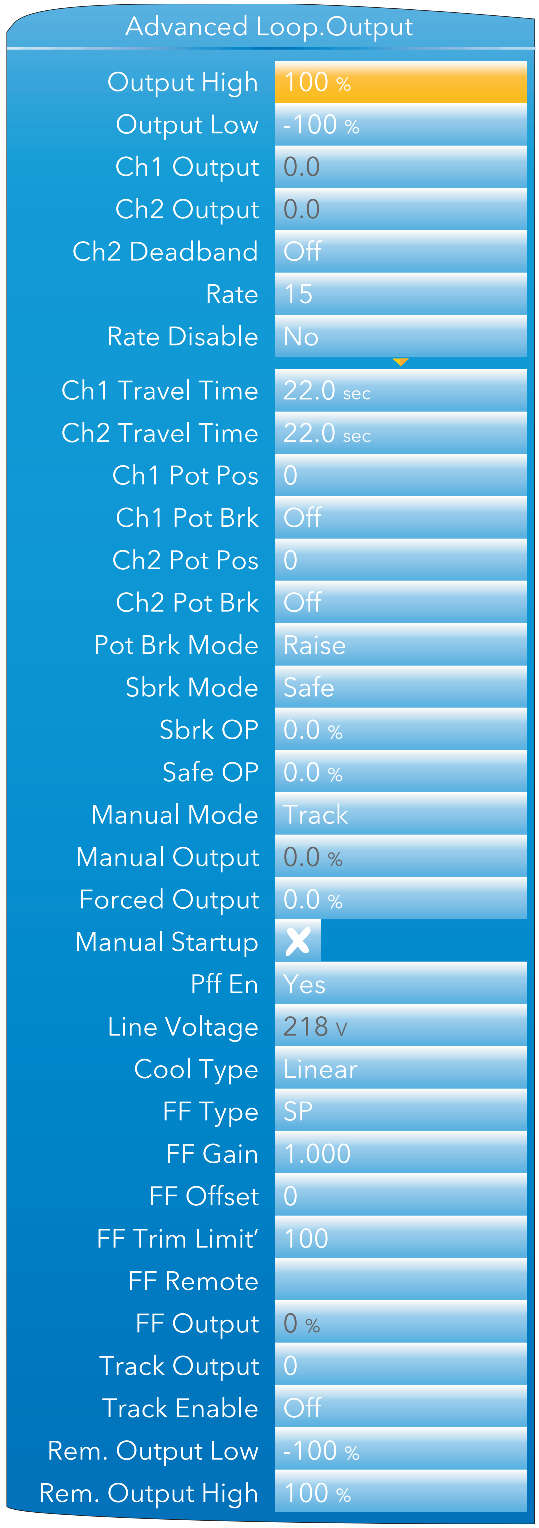



Figure 4.7.8 Advanced Loop Output menu

### 4.7.8 ADVANCED LOOP OUTPUT MENU (Cont.)

| | |
|---|---|
| Output High | The maximum output power to be delivered by channels 1 and 2, where 100% is full power. The valid input range is Output Low to 100.0%. Reducing this value reduces the rate of change of the process, but it also reduces the controller's ability to react to perturbations and can even cause it to fail to achieve setpoint. |
| Output Low | The minimum power, or the maximum 'negative' (i.e. cooling) power to be delivered by the system. |
| Ch1 Output | Displays the positive power values used by the heat output. |
| Ch2 Output | Displays the cooling power values for channel two. Appears as a value between Output High and -100%, where -100% represents full cooling power. |
| Ch2 Deadband | A gap (in %) between output 1 switching off, and output 2 switching on, and *vice-versa*. Valid inputs are 0 (off) to 100%. |
| Rate | Limit on the rate at which the output from the PID can change. Can be useful in preventing rapid changes in output that could damage the process, heater elements etc. |
| Rate Disable | The Output Rate limit may be disabled by setting its value to 0.0. Alternatively, for some applications it is useful to be able to wire to the Output Rate Disable so that 'Rate' can be switched on/off during stages of the process. For example, Rate Disable can be used with the programmer event outputs to control the output rate of change during a particular segment. |
| Ch1 Travel Time | Appears only if Setup menu parameter 'Slave Channel 1' is set to 'VPB'. This is the valve travel time from closed (0%) to open (100%). In a valve positioning application, channel 1 output is connected by a single software 'wire' to a Valve Raise/Valve Lower relay pair. For heat/cool applications, channel 1 is associated with the heating valve. Valid entries: 0.0 to 1000.0 seconds. |
| Ch2 Travel Time | Appears only if Setup menu parameter 'Slave Channel 2' is set to 'VPB' . This is the valve travel time from closed (0%) to open (100%). For heat/cool applications, channel 2 is associated with the cooling valve. Valid entries: 0.0 to 1000.0 seconds. |
| Ch1 Pot Pos* | The position of the channel one actuator as measured by the feedback potentiometer. |
| Ch1 Pot Brk* | 'On' indicates that the input to the relevant channel is open circuit. |
| Ch2 Pot Pos* | The position of the channel two actuator as measured by the feedback potentiometer. |
| Ch2 Pot Brk* | 'On' indicates that the input to the relevant channel is open circuit. |
| Pot Brk Mode* | Defines the action to be taken if a potentiometer break is detected:<br>Raise: opens the valve<br>Lower: closes the valve<br>Rest: the valve remains in its current state.<br>Model: the controller tracks the position of the valve and sets up a model of the system so that it continues to function if the potentiometer becomes faulty. This does not mean that the potentiometer can be omitted with VPB, as the accuracy of valve position control is reduced without it. |

* Note: These parameters appear only if the 'Setup' menu parameter 'Slave Channel 1' or 'Slave Channel 2' (as appropriate) is set to 'VPB'. The Setup menu is described in section 4.7.2.

| | |
|---|---|
| SBrk Mode | Defines the action to be taken in the event of a sensor break.<br>Safe: The output adopts the value configured in 'Sbrk OP', below.<br>Hold: The output remains at its current level. |
| Sbrk OP | The value to be output if a Slave sensor break occurs, and SBrk Mode (above) is set to 'Safe'. |
| Safe OP | The output level adopted when the loop is inhibited (Main menu section 4.7.1). |

### 4.7.8 ADVANCED LOOP OUTPUT MENU (Cont.)

| | |
|---|---|
| Manual Mode | Selects the type of transition to occur when changing to manual cascade mode (section 4.7.1):<br>Track: Whilst in Auto mode, the manual output tracks the control output so that there is no change of output when manual mode is switched to.<br>Step: On transition to manual mode, the output is set to the value entered for 'Forced-OP' (below).<br>Last Man. Out: On transition to manual mode, the output adopts the manual output value as last set by the operator. |
| Manual Output | The output when the loop is in manual mode. In manual mode the controller limits the maximum power, but it is not recommended that it be left unattended at high power settings. It is important that over range alarms are fitted to protect the process. |

Note: It is recommended that all processes are fitted with an independent over range detection system.

| | |
|---|---|
| Forced Output | Forced Manual output value. When 'Manual Mode' = 'Step', this is the output value adopted when changing from Auto to Manual mode. |
| Manual Startup | When set to off (cross symbol), the controller powers up in the same (auto or manual) mode that obtained when it was switched off. When set to on (tick symbol) the controller always powers up in manual mode. |
| Pff En | Power feed forward enable. 'Yes' enables power feed forward (adjusts the output signal to compensate for variations is supply voltage. 'No' disables Pff. See section B2.6.6 for further details. |
| Line Voltage | Read only display of the current supply voltage. |
| Cool Type | Appears only if 'Ch2 Control' = 'PID' in the setup menu (section 4.7.2) and allows the user to enter the appropriate type of cooling (section B2.6.7):<br>Linear: For use when controller output changes linearly with PID demand.<br>Oil: For oil cooled applications<br>Water: For water cooled applications<br>Fan: For forced air cooling. |
| FF Type | Feed forward type (section B2.6.8):<br>None: No signal fed forward.<br>Remote: A remote signal is fed forward.<br>SP: Setpoint is fed forward.<br>PV: PV is fed forward. |
| FF Gain | For FF types 'PV' and 'SP', this scales the feed forward signal. |
| FF Offset | For FF types 'PV' and 'SP', this defines the offset of the scaled feed forward signal. |
| FF Trim Limit | For FF types 'PV' and 'SP', defines symmetrical limits about the PID output which are applied to the scaled feed forward signal. |
| FF Remote | Allows another value from the strategy to be used as the primary control variable in the feed forward strategy. The gain and offset are not applied to the remote value. |
| FF Output | For FF types 'PV' and 'SP', this is the calculated (scaled, offset and trimmed) feed forward signal. FF OP = FF gain (input + FF Offset) |
| Track Output | If 'Track Enable' (below) is set to 'Yes', this is the value for the loop output to track when output track is enabled. |
| Track Enable | When set to 'Yes', the output follows the Track OP value (above). When subsequently set to 'Off' the loop makes a bumpless return to control. |
| Rem. Output Low/High | Used to limit the output when using a remote source. These limits cannot exceed the 'Output Low' and 'Output High' values described earlier in this section. |

### 4.7.9 Advanced Loop Diagnostics menu

| | |
|---|---|
| Master Error | The difference in value between the setpoint and the PV for the Master (Read only). |
| Slave Error | The difference in value between the setpoint and the PV for the Slave (Read only). |
| (M)Prop. Output | Shows the proportional term contribution to the control output of the Master (Read only). |
| (M)Integral Out | Shows the integral term contribution to the control output of the Master (Read only). |
| (M)Deriv. Output | Shows the derivative term contribution to the control output of the Master (Read only). |
| (S)Prop. Output | Shows the proportional term contribution to the control output of the Slave (Read only). |
| (S)Integral Out | Shows the integral term contribution to the control output of the Slave (Read only). |
| (S)Deriv. Output | Shows the derivative term contribution to the control output of the Slave (Read only). |
| Target Output | The requested control output. The target of the active output if rate limiting is active. (Read only.) |
| Loop Break (S) | Loop Break Alarm (Read only). Becomes active 'Yes' if the relevant loop break time (LBT1/2/3), set in the Slave PID menu (section 4.7.5) is exceeded, otherwise 'No' is displayed. |
| Loop Break (M) | Loop Break Alarm (Read only). Becomes active 'Yes' if the Master loop break time (LBT), set in the Master PID menu (section 4.7.4) is exceeded, otherwise 'No' is displayed. |
| Sensor Break (S) | Indicates Slave sensor break status (Read only). On (tick symbol) indicates a sensor break has occurred; Off (cross symbol) shows that no sensor breaks have been detected. |
| Sensor Break (M) | Indicates Master sensor break status (Read only). On (tick symbol) indicates a sensor break has occurred; Off (cross symbol) shows that no sensor breaks have been detected. |
| Sched PB | The scheduled proportional band for the current PID set. |
| Sched Ti | The scheduled integral time for the current PID set. |
| Sched Td | The scheduled derivative time for the current PID set. |
| Sched CBH | The scheduled cutback high value for the current PID set. |
| Sched CBL | The scheduled cutback low value for the current PID set. |
| Sched MR | The scheduled manual reset value for the current PID set. |
| Sched Loop Break | The scheduled loop break time for the current PID set. |
| Sched R2G | The scheduled relative cool gain value for the current PID set. |
| Sched Out High | The scheduled output high limit for the current PID set. |
| Sched Out Low | The scheduled output low limit for the current PID set. |
| Working Out Low | The low limit for the working output (Read only). This is the value used to limit the output power of the loop and is derived from the gain scheduled limit, the remote limit and the safety limit. |
| Working Out High | The high limit for the working output (Read only). This is the value used to limit the output power of the loop and is derived from the gain scheduled limit, the remote limit and the safety limit. |
| Master FB | Master FB is the value of the master control output after limiting and is used for Integral desaturation. |
| Calc OP | Master P+I+D |
| HiSatLim | HiSatLim is an internally generated limit |
| LoSatLim | LoSatLim is an internally generated limit |
| OPPID | Master control output It will be the same as Calc OP if the master is not in Cutback |

## 4.8 PROGRAMMER CONFIGURATION

The programmer option allows the user to configure a setpoint program with one or two channels, as required. The program can be run from the Programmer operator display page (section 3.4.9) or can be controlled by inputs received from other parameters. In particular, the programmer is intended for use with the loop or advanced loop options.

The programmer configuration is separated into a number of areas as depicted in the overview below. The segment configuration (ramp type etc.) is carried out from the programmer edit page, also described in section 3.4.9.

Programmer

| Features (Section 4.8.1) | FTP (Section 4.8.2) | Set Up (Section 4.8.3) | Run (Section 4.8.4) |
|---|---|---|---|
| Holdback | IP Address | Channels | Mode |
| PV Event | Username | Prog Mode Access | Status |
| User Value | password | Prog Edit Access | Ch1(2) PSP |
| Messages | | Prog Store Access | Segment Name |
| FTP Store | | Rate Resolution | Segment type |
| | | Ch1(2) Units | Segment Time Left |
| | | Ch1(2) Resolution | Segment Time Run |
| | | Ch1(2) Servo To | Prog Time Left |
| | | Ch1(2) PV Input | Prog Time Spent |
| | | Ch1(2) SP Input | Prog Time Run |
| | | Power Fail Action | Duration |
| | | Max Events | Ch1(2) TSP |
| | | Reset Events 1 to 8 | Ch1(2) rate |
| | | Reset Ch1(2) UV | Ch1(2) Time |
| | | Run | Cycles Left |
| | | Hold | Ch1(2) User Val |
| | | Reset | Ch1(2) PV Event |
| | | Run reset | Event 1 to 8 |
| | | Run Hold | End Output |
| | | Wait Digital | |
| | | Wait analogue 1(2) | |
| | | Advance | |
| | | Operation | |
| | | Status | |

### 4.8.1 Programmer Features menu

This menu allows the user to enable/disable some of the items presented to the user in the Programmer edit page described in section 3.4.9. Features are enabled/disabled by using the up/down arrow keys to highlight the required item and then using the scroll button to toggle between enabled (tick) and disabled (cross). Typically, items would be left disabled in order to reduce the number of configuration fields presented to a user who may not need all such features.

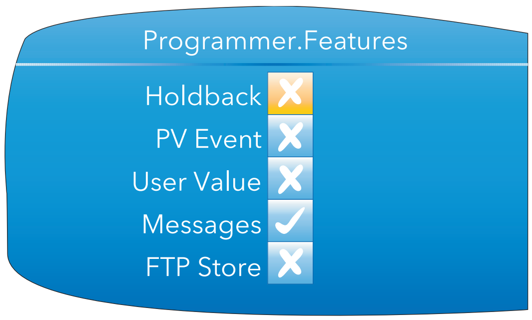



Figure 4.8.1 Programmer features menu

| | |
|---|---|
| Holdback | Holdback pauses the program (freezes the Programmer setpoint (PSP) and the time remaining parameters) if the difference between the Process value (PV) and the PSP exceeds a user-specified amount (deviation). The program remains paused until the PV returns to within the specified deviation.<br>In ramp or step segments, holdback indicates that the PV is lagging the SP by more than the specified amount and that the program is waiting for the process to catch up. In a dwell segment, holdback is used to guarantee that a work piece stays at set-point within a specified tolerance for the specified dwell duration.<br>Holdback type and deviation value are configured, on a per program basis, to be applied to either the entire program or to individual segments. See Program edit (section 3.4.9) for details. |
| PV Event | A PV Event is available for each channel in every segment except for Wait and Go Back segment types. A PV Event is an absolute or deviation analogue alarm on the channel PV, and can be used to trigger a secondary process, or to trigger an analogue alarm. |
| User Value | A user value can be entered for every segment (except for Wait or Go Back types) and when the segment is entered, this value is transferred to the associated User Value Output parameter, which could be wired to another parameter to form part of an application strategy. |
| Messages | Table 4.8.1, below, lists the programmer specific events that generate messages that are displayed in the message summary and recorded into the history file.<br>It is also possible to trigger custom messages from any of the programmer outputs via user wiring. The program name and segment name can be embedded in custom messages by inserting the modbus address for the current program / segment name parameters in square brackets i.e.:<br>[<current_program_name_modbus_address>]<br>[<current_segment_name_modbus_address>] |

### 4.8.1 PROGRAMMER FEATURES CONFIGURATION (Cont.)

| Event | Message |
|---|---|
| Program Run | <program_name>: Run |
| Program End | <program_name>: Complete |
| Program Hold | <program_name>: <segment_name>: Hold |
| Program Resume | <program_name>: <segment_name>: Resume |
| Program Reset | <program_name>: <segment_name>: Reset |
| Segment Start | <program_name>: <segment_name>: Segment Start |
| Advance | <program_name>: <segment_name>: Advanced |
| Holdback | <program_name>: <segment_name>: Holdback:Channel No. |
| PV Event | <program_name>: <segment_name>: PV Event:Channel No. |

Table 4.8.1 Programmer messages

FTP Store — If this feature is enabled, an FTP menu item appears in the top level Programmer configuration menu. 'FTP' allows the user to enter communications parameters for the host computer which is to act as the ftp server.
FTP Store allows the user to set-up a centralised program store from which several instruments can select their program.

Notes

1 A maximum of 100 entries is supported on all drives. Directory trees are supported for both USB and FTP, and if the root of the drive contains only files (no directories), then up to 100 files are listed. If the root of the drive contains directories then each directory can contain 100 entries (but one of these entries will be taken up by '..' to return to the directory above).
2 Program files are in compressed XML (.uipz) file format.
3 When a program is selected from an FTP server a local copy of the program file is made within the instrument before being processed. It should be noted that the number of program files in the internal 'user' drive does not affect this local copy, and therefore a program from an FTP server can be loaded, even when the internal user drive is full.
4 As the loaded program resides in the current program database it is automatically included in a clone file. In addition, program files stored in the internal program drive are included in a clone file (refer to 'Cloning', below).
5. On the internal program drive only a flat directory structure is supported. However, full tree directories are supported on both the USB memory stick and FTP server (accessed via the HMI File Explorer).
6. It is not possible to store program files on an external device. Programs selected from an external device can, however, be stored in the internal program store.
7. It is not possible to select a program from an external device over comms and iTools.

### CLONING

Each program file stored locally on the instrument IS included in a clone file as a Binary Large Object (BLOB), similar to the Graphical Wiring Editor layout. Each program file BLOB contains the program filename.

When loading a clone file, existing programs in the instrument's internal drive are deleted, and program file BLOB(s) in the clone are reformatted into program files by the instrument.

### 4.8.2 Programmer FTP menu

Note: This menu item is accessible only if 'FTP' has been enabled in the Programmer features menu described above.

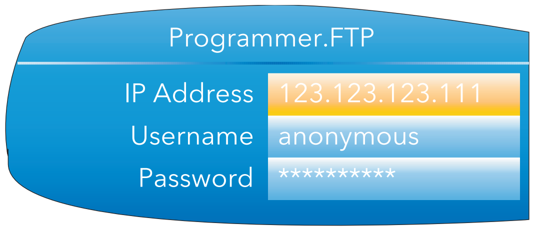



Figure 4.8.2 Programmer FTP menu

| | |
|---|---|
| IP Address | The IP address of the FTP server. |
| Username | The User Name entered when setting up the FTP server |
| Password | The password associated with the above User Name. |

Section C2 gives an example of how to set up an FTP server using 'Filezilla'

### 4.8.3 Programmer Setup menu

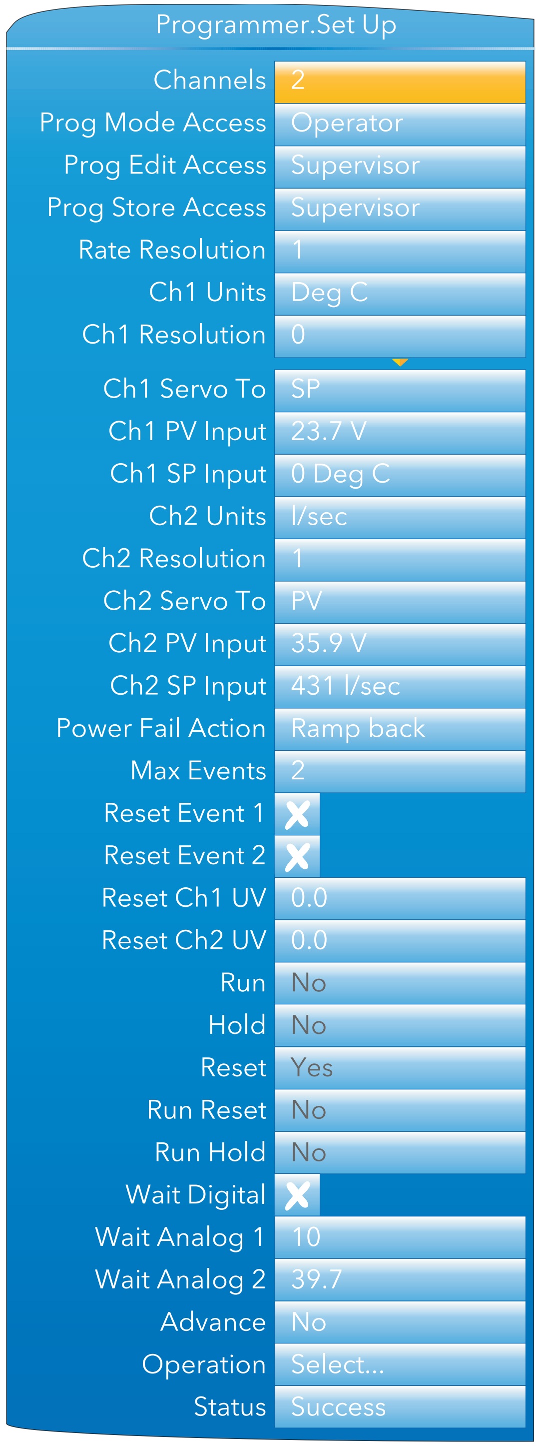



Figure 4.8.3 Programmer Set Up menu

| | |
|---|---|
| Channels | The number of channels to be profiled. 1 = single channel mode, 2 = dual channel sync-all mode |
| Prog Mode Access | Sets the minimum access level (Logged off, Operator, Supervisor) for allowing changes to the current program mode (run, hold or reset) |

### 4.8.3 PROGRAMMER SET UP MENU (Cont.)

| | |
|---|---|
| Prog Edit Access | Sets the minimum access level (Logged off, Operator, Supervisor, Engineer) for loading programs, and for allowing edits to the current program including permission to advance a segment. |
| Prog Store Access | Sets the minimum access level (Logged off, Operator, Supervisor, Engineer) that allows users to copy, store and delete programs. |
| Rate Resolution | Sets the resolution (0 to 4 decimal places) of ramp rates when read from / written to via scaled integer comms. |
| Ch1 Units | Five-character (max.) descriptor for channel 1 units. If wired, the units will be those of the wire source. |
| Ch1 Resolution | Number of decimal places for channel 1 value. If wired, the value will be that of the wire source. |
| Ch1 Servo To | Determines whether the programmer starts running channel 1 from the control loop's configured set-point (servo to SP), or from the current process value (servo to PV). |
| Ch1 PV Input | Various programmer functions (for example Ch1 Servo to PV), require the PV value of the loop that the programmer is trying to control. The parameter is normally wired from the loop's Track PV parameter. |
| Ch1 SP Input | Various programmer functions (for example Ch1 Servo to SP), require the SP value of the loop that the programmer is trying to control - it is normally wired from the loop's Track SP parameter. |
| Ch2 Units | As 'Ch1 Units', above but for channel 2. Appears only if 'Channels' is set to '2'. |
| Ch2 Resolution | As 'Ch1 Resolution', above, but for channel 2. Appears only if 'Channels' is set to '2'. |
| Ch2 Servo To | As 'Ch1 Servo To', above, but for channel 2. Appears only if 'Channels' is set to '2'. |
| Ch2 PV Input | As 'Ch1 PV Input', above, but for channel 2. Appears only if 'Channels' is set to '2'. |
| Ch1 SP Input | As 'Ch1 SP Input', above, but for channel 2. Appears only if 'Channels' is set to '2'. |
| Power Fail Action | If the power supplied to the instrument is interrupted, the program status is retained and when power is restored, the instrument performs the selected power fail action.<br>Continue: The programmer set-point returns immediately to its last value prior to the power down and the program continue to run from that point.<br>Reset: The program resets.<br>Ramp Back: The programmer servos the programmer set-point to the channel PV, and ramps to the target set-point at the rate prior to the power-fail. The time remaining for the segment is recalculated. |

Notes:

1. If the interrupted segment was a 'time to target' ramp, then when power is returned the calculated ramp rate prior to the interruption is used.
2. If the interrupted segment was 'Dwell', then the ramp rate is determined by the previous ramp segment. On achieving the dwell set-point, the dwell period continues.
3. If a previous ramp segment does not exist (i.e. the first segment of a program is a dwell), the dwell continues at the 'servo to PV' programmer set-point.

| | |
|---|---|
| Max Events | Configures the maximum number of event outputs (0 to 8). |
| Reset Event N | Sets the state of event output 'N' when the program is in reset. Appears only if 'Max Events' is > (N-1). |
| Reset Ch1 UV | Enter the value to be written to user value 1 when the program is in reset. Appears only if 'User Value' feature is enabled in Programmer.Features configuration (section 4.8.1). |
| Reset Ch2 UV | Enter the value to be written to user value 2 when the program is in reset. Appears only if 'User Value' feature is enabled in Programmer.Features configuration (section 4.8.1) and 'Channels' = '2' |
| Run | The input that causes the programmer to place the current program in Run mode. |
| Hold | The input that causes the programmer to place the current program in Hold mode. |

### 4.8.3 PROGRAMMER SET UP MENU (Cont.)

| | |
|---|---|
| Reset | The input that causes the programmer to place the current program in Reset mode. |
| Run Reset | Dual functionality input, that causes the programmer to place the current program in Run or Reset mode. |
| Run Hold | Dual functionality input, that causes the programmer to place the current program in Run or Hold mode. |
| Wait Digital | The Boolean input that is used in Wait segments. |
| Wait Analog 1 | The analogue input associated with channel 1 that is used in wait segments. |
| Wait Analog 2 | The analogue input associated with channel 2 that is used in wait segments. Appears only if 'Channels' = '2' |
| Advance | The input to advance the current segment |
| Operation | Program file operation selection parameter. See 'Program editing' (section 3.4.9) for further details |
| Status | Status indication of the selected file operation.See 'Program editing' (section 3.4.9) for further details |
| Amended | Indicates whether the current program has been amended since being loaded (Comms only) |
| File Error Status | File operation error status (Busy, OK, Load Open File Error, Store Open File Error, Delete File Fail, Copy File Fail, Invalid Format, Invalid Device, Invalid Version, Invalid Num Channels, Parameter Write Fail, Store Operation Did Not Complete, Load Operation Did Not Complete, Delete Operation Did Not Complete, Copy Operation Did Not Complete, Invalid Filename, Unspecified Error). Available only over Comms as the error is displayed on the display screen.<br>'Parameter Write Fail' indicates that one or more program/segment parameters failed to be written to during a 'Load' operation. This is generally caused by a program that contains features (i.e. Holdback, User Values, PV Events) which are disabled in the instrument's Programmer block, or the program contains more Event Outputs than configured in the instrument's Programmer block. |

### 4.8.4 Programmer Run menu


Figure 4.8.4 Programmer Run menu

| | |
|---|---|
| Mode | Current program mode (Run, Hold, Reset). |
| Status | Current program status (Running, Holding, Holdback, Waiting, Reset, Complete) |
| Ch1 PSP | The output setpoint for channel 1. |
| Ch2 PSP | The output setpoint for channel 2. Appears only if 'Channels' = '2' in the Set Up menu (section 4.8.3). |
| Segment | Name of the current segment as entered in the Program Edit page (section 3.4.9) |
| Segment Type | Current segment type as entered in the Program Edit page (section 3.4.9) |
| Seg Time Left | Indicates the minimum amount of time left in the current segment. |
| Seg Time Run | The length of time that the current segment has been running. This value does not include time spent in Hold, Holdback or Waiting |
| Prog Time Left | Shows the minimum amount of time left before the program completes. Each segment can be up to 500 hours in length. The maximum display is 500 hours, and if the length of the entire program is greater than this, the display remains at 500 until the remaining time falls below 500 hours. |
| Prog Time Spent | Indicates the length of time the current program has been running, including time spent in Hold, Holdback or Waiting |

### 4.8.4 PROGRAMMER RUN MENU (Cont.)

| | |
|---|---|
| Prog Time Run | The length of time the current program has been running. This value does not include time spent in Hold, Holdback or Waiting |
| Duration | For Dwell segments only, this is the dwell duration. |
| Ch1 TSP | For Ramp and Step segments, this is the current target setpoint for channel 1. |
| Ch1 Time | For Ramp segments, this is the configured time for channel 1 to reach it's Target Setpoint (TSP) |
| Ch2 TSP | For Ramp and Step segments, this is the current target setpoint for channel 2. Appears only if 'Channels' = '2' in the Programmer Set Up menu (section 4.8.3). |
| Ch2 Time | For Ramp segments, this is the configured time for channel 2 to reach it's Target Setpoint (TSP). Appears only if 'Channels' = '2' in the Programmer Set Up menu (section 4.8.3) |
| Cycles Left | The number of Go Back cycles remaining before the Go Back loop ends. |
| Ch1 User Val | The value of user value 1 in the current segment. Appears only if the 'User Value' feature is enabled in the Programmer Features menu (section 4.8.1). |
| Ch2 User Val | The value of user value 2 in the current segment. Appears only if the 'User Value' feature is enabled in the Programmer Features menu (section 4.8.1) and if 'Channels' = '2' in the Programmer Set Up menu (section 4.8.3) |
| Ch1 PV Event | The state of channel 1 PV event (Off = Cross symbol, On = Tick). Appears only if the 'PV Event' feature is enabled |
| Ch2 PV Event | The state of channel 2 PV event (Off = Cross symbol, On = Tick). Appears only if the 'PV Event' feature is enabled and if 'Channels' = '2' in the Programmer Set Up menu (section 4.8.3). |
| Event 1 to 8 | The state of event output 1 to 8 for the current segment (Off = Cross symbol, On = Tick). The number of events appearing is defined in the Programmer Set Up menu (section 4.8.3) (Max Events) |
| End Output | The output that is set by the end segment (Off = Cross symbol, On = Tick). |

### 4.8.5 Connecting the programmer to a loop

Below are some examples of how programmers and loops may be soft-wired together so that the programmer has access to the Loop PV and Loop setpoint. The examples are taken from iTools (section 6), but may be carried out through User Wiring (section 7) if more convenient.

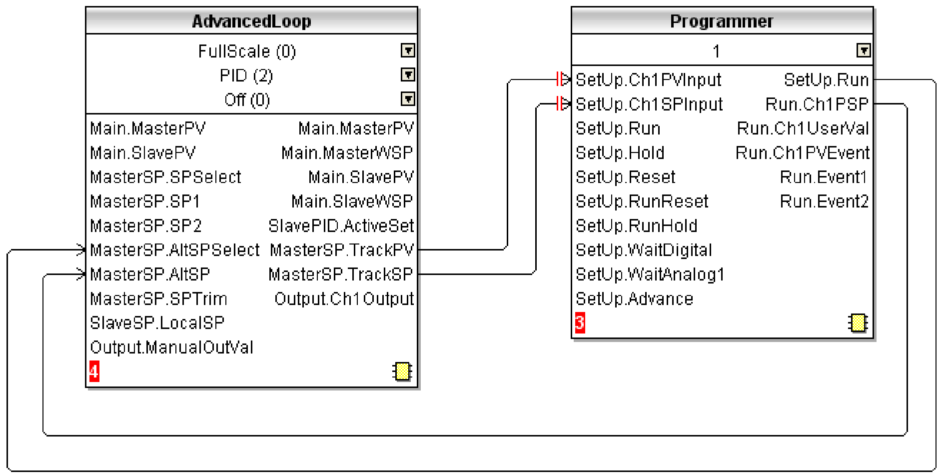


Figure 4.8.5a Advanced loop to Programmer basic wiring

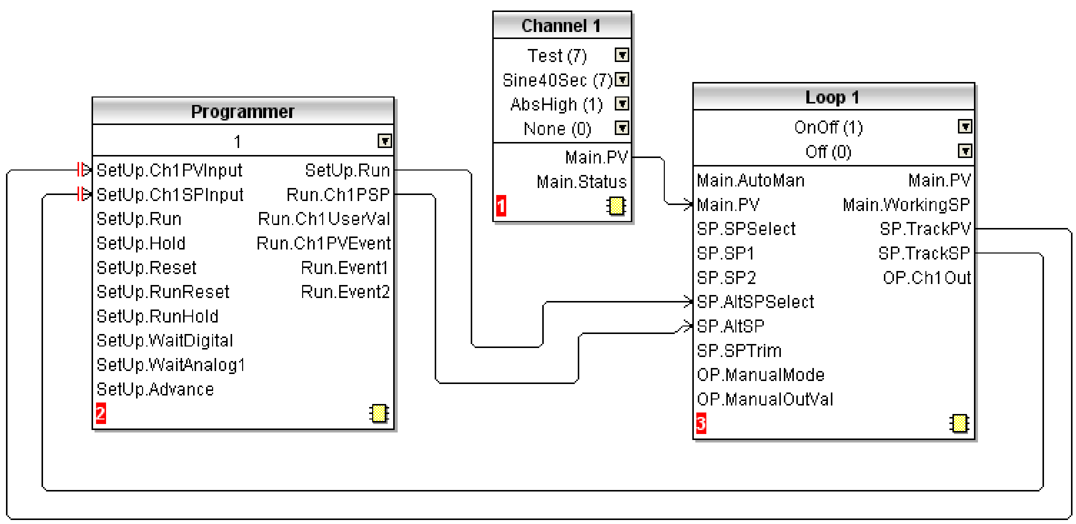


Figure 4.8.5b Programmer to Loop basic wiring

### 4.8.5 CONNECTING THE PROGRAMMER TO A LOOP (Cont.)

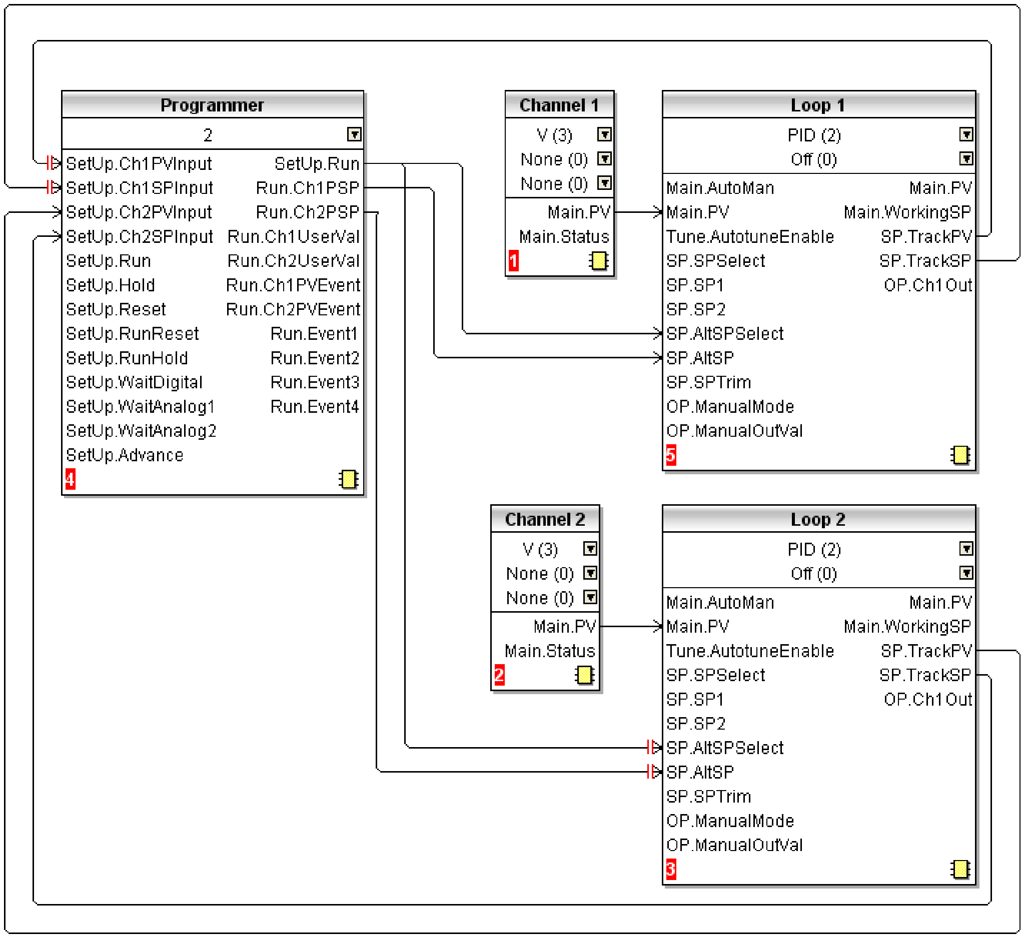



Figure 4.8.5c Dual programmer to two loops basic wiring

### 4.8.6 Configuration by Modbus Comms

It is possible to configure, store, delete, or load a program via Modbus comms by setting the Program and Segment parameters using either their scaled integer or native modbus addresses (section 5.3).

**EXAMPLE 1: CONFIGURE A PROGRAM**

To configure a simple Ramp-Dwell-Ramp program via modbus comms:

Set Segment.1.Type (address 15040) to Ramp (1)

Set Segment.1.Ch1TSP (address 15042) to 60.0 (600 - 1dp)

Set Segment.1.Ch1Time (address 15044) to 60s (60s)

Set Segment.2.Type (address 15088) to Dwell (2)

Set Segment.2.Duration (address 15089) to 120s (120)

Set Segment.3.Type (address 15136) to Ramp (1)

Set Segment.3.Ch1TSP (address 15138) to 0.0 (0 - 1dp)

Set Segment.3.Ch1Time (address 15140) to 180s (180)

**EXAMPLE 2: STORE A PROGRAM**

To store the current program:

Set Programmer.FileList.FilenameEntry (address 27281) to required filename (e.g. George)

Set Programmer.Setup.Operation (address 14912) to Store (4)

Read Programmer.Setup.Operation (address 14912) until it returns Select (1)

Read Programmer.Setup.Status (address 14913) to get the status of the store operation (Success = 1, Failed = 2)

**EXAMPLE 3 LIST STORED PROGRAMS**

To get a listing of stored program files:

Set Programmer.FileList.Operation (address 14976) to Get Listing (1)

Read Programmer.FileList.Operation (address 14976) until it returns Complete (0)

Read Programmer.FileList.Filename1 to 100 parameters (address 30976 - 31075)

Note: for each filename parameter perform a 21 register block read starting from the base address of the parameter, 1st null string indicates end of List.

**EXAMPLE 4: LOADING PROGRAMS**

To load a program:

Get a listing as described above

Set Programmer.FileList.FilenameEntry (address 27281) to the filename to be loaded (e.g. George)

Set Programmer.Setup.Operation (address 14912) to Load (2)

Read Programmer.Setup.Operation (address 14912) until it returns Select (1)

Read Programmer.Setup.Status (address 14913) to get the status of the store operation (Success = 1, Failed = 2)

## 4.9 MODBUS MASTER CONFIGURATION

Modbus master configuration is divided into two areas: a) setting up the slave(s), including diagnostics, and b) defining the locations of the parameters to be read. Figure 4.9 shows an overview.

Section 3.4.12 shows the Modbus Master display page, and describes the configuration options available there.

Note: Versions 2.40 to 2.50 of the Mini8 Controller, and versions 2.70 to 3.20 of the Model 3550 controller are supported. It is not guaranteed that later software versions of these instruments will be fully compatible.

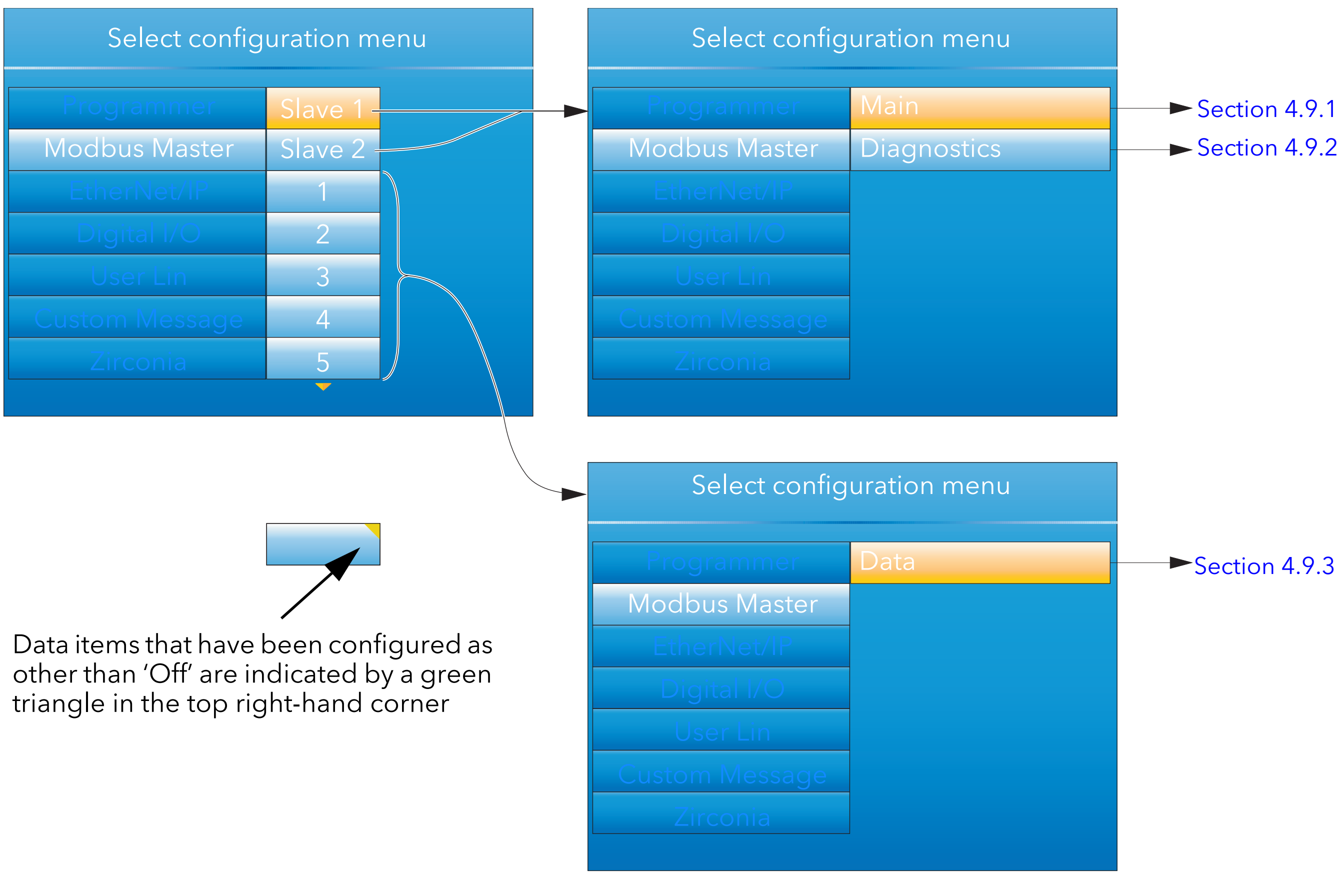
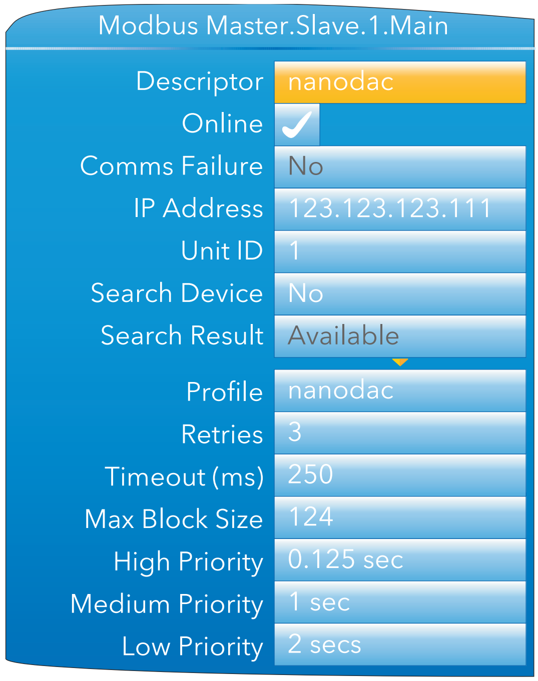



Figure 4.9 Modbus Master configuration top level menus

### 4.9.1 Slave Main menu

This allows the IP address, Unit ID and other communications parameters to be entered for Slaves 1 and 2.




| Available Priority selections | | |
|---|---|---|
| Seconds | Minutes | Hours |
| 0.125 | 1 | 1 |
| 0.25 | 2 | |
| 0.5 | 5 | |
| 1 | 10 | |
| 2 | 20 | |
| 5 | 30 | |
| 10 | | |
| 20 | | |
| 30 | | |

Figure 4.9.1 Modbus Master Slave 1 configuration (Slave 2 similar)

| | |
|---|---|
| Descriptor | A descriptor for this instrument. For use in Modbus communications, this is not the same as the 'Name' which appears in the Instrument Info configuration (section 4.14). |
| Online | Disabled by default (Cross symbol). Must be enabled (highlighted using the down arrow, then edited by the scroll button) to allow the remaining configuration items to appear and to allow data transactions be processed. Setting the slave offline temporarily disables data transactions - it does not reconfigure them. |
| Comms Failure | Active (yes) if a data item has failed to respond after all retries. |
| IP Address | The IP address of the Slave device |
| Unit ID | The Unit Id or Modbus address to use in each data transaction with the slave device. Limits are 1 to 255 |
| Search Device | Setting this to 'Yes' searches the network to see if the device with the specified IP address and Unit ID is available. If so, the descriptor will be overwritten to indicate what type of device has been found. |
| Search Result | The status of the selected 'Search Device' request (Searching, Available, Unreachable). Search activity is indicated by a rotating animated display in the 'Searching' field. |
| Profile | A number of profiles are held within the instrument that match a selection of known devices. If the device is 'known', its type, model number etc. is displayed. If the device is unknown, '3rd Party' appears instead. |
| Retries | The number of times (0 to 3) to re-send a data transaction to the device if no response is received within the configured timeout period (below). |
| Timeout | The timeout period for each Modbus transaction in ms |
| Max Block Size | The maximum number of registers (16bit words) that a single data transaction is permitted to contain |
| High Priority | The interval rate between each high priority data transaction. Default = 0.125 second. |
| Medium Priority | The interval rate between each medium priority data transaction. Default = 1 second. |
| Low Priority | The interval rate between each low priority data transaction. Default = 2 seconds. |

### 4.9.1 SLAVE MAIN MENU (Cont.)

#### PRIORITY LEVELS

Three levels of update rate can be entered for use in data configuration (section 4.9.3), to define how often a value is read or written. In order to optimise performance, it is recommended that the slowest rate consistent with requirements be selected. The intervals are selected from a scroll list see figure 4.9.1 above.

## 4.9.2 Slave Diagnostics menu

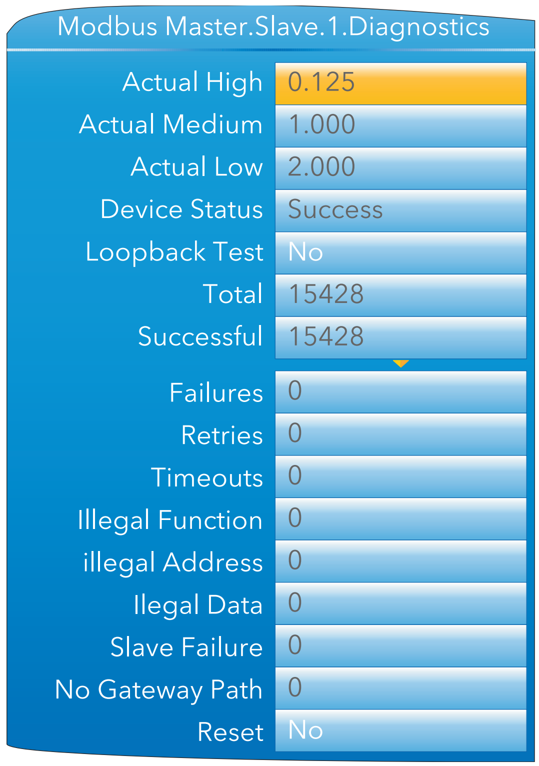



Figure 4.9.2 Diagnostics menu

Note: Diagnostic values are reset on power up

| | |
|---|---|
| Actual High | The high priority rate that this slave is actually running at. This can never be faster than the high priority rate that was configured for this device (Slave Main menu, above), but if the master is heavily loaded the rate may be lower than that specified. |
| Actual Medium | The medium priority rate that this slave is running at. This can never be faster than the medium priority rate that was configured for this device (Slave Main menu, above), but if the master is heavily loaded the rate may be lower than that specified. |
| Actual Low | The actual low priority rate that this slave is running at. This can never be faster than the low priority rate that was configured for this device (Slave Main menu, above), but if the master is heavily loaded the rate may be lower than that specified. |
| Device Status | The status of the last transaction to this slave<br>Success: The transaction was successfully actioned by the slave device<br>Timeout: There was no response from the slave device to a given request within the configured time<br>Illegal Address: The request to the slave device contained an invalid modbus address. The address may be for a read only parameter<br>Illegal Value: The request to the slave device contained invalid data for the specified parameter<br>Bad Sub: The sub function code in the request was invalid |

### 4.9.2 SLAVE DIAGNOSTICS MENU (Cont.)

**DEVICE STATUS (Cont.)**

| | |
|---|---|
| | Idle: This data item is currently idle and not communicating with the slave device |
| | Illegal Code: The slave does not support the function code transmitted by the master. |
| | Pending: The request is waiting to be sent, the most likely cause being that the slave device has not been set to online |
| Loopback Test | If set to 'Yes', Sends a function code 8 transaction to the slave, and waits for a response. |
| Total | A count of all the transactions sent to the slave including reads, writes both good and failed transactions. |
| Successful | A count of all the successful transactions sent to the slave. |
| Failures | A count of all the unsuccessful (failed) transactions sent to the slave. May be caused by Illegal Function, Illegal Address etc. failures, as detailed below |
| Retries | The number of transactions that were re-sent because of timed out responses from the slave devices. |
| Timeouts | A count of all the transactions sent to the slave for which no response was received within the configured timeout period. |
| Illegal Function | A count of all the transactions sent to the slave that the slave claimed contained an invalid function code. Exception code (1). |
| Illegal Address | A count of all the transactions sent to the slave that the slave claimed contained an invalid Modbus register address. Exception code (2). |
| Illegal Data | A count of all the transactions sent to the slave that the slave claimed contained an invalid value. Exception code (3) |
| Slave Failure | A count of all the times this slave device has failed to communicate. Exception code (4) |
| No Gateway Path | A count of all the times it has not been possible to access the slave device as it is on another network that requires a gateway for access |
| Master Rejects | A count of all the transactions that the Modbus Master has refused to send to the slave due to invalid configuration data |
| Reset | A one shot action that immediately resets all diagnostics counts. |

### 4.9.3 Modbus master data configuration

This is the area of configuration in which the individual data items are selected for transmission across the Modbus master communications link. The configuration fields that appear depends on the parameter selected, so the examples given here will probably not match those that appear to the user. The parameters that appear in the 'parameter List' scroll menu depends on the slave model.

**EXAMPLE 1: TARGET SP1 WITH NANODAC SLAVE**

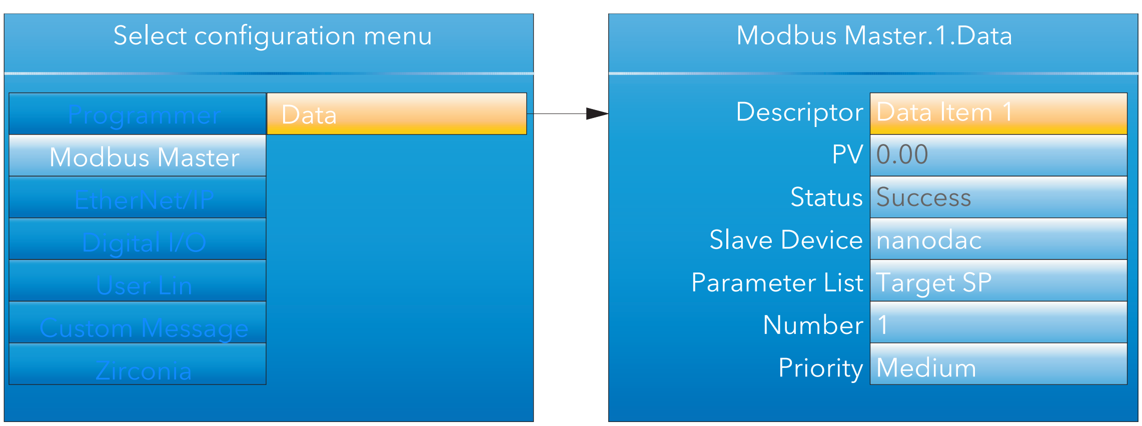



Figure 4.9.3a Target Setpoint

### 4.9.3 MODBUS MASTER DATA CONFIGURATION (Cont.)

#### EXAMPLE 2 USER DEFINED PARAMETER

This allows the user to enter a Modbus address (decimal) and a data type in order to read the value of a parameter from or write a parameter value to the slave. Modbus address and data types must be obtained from the documentation supplied with the slave device. For convenience, this example uses a nanodac as the slave; table 5.3 of this document providing the required data.

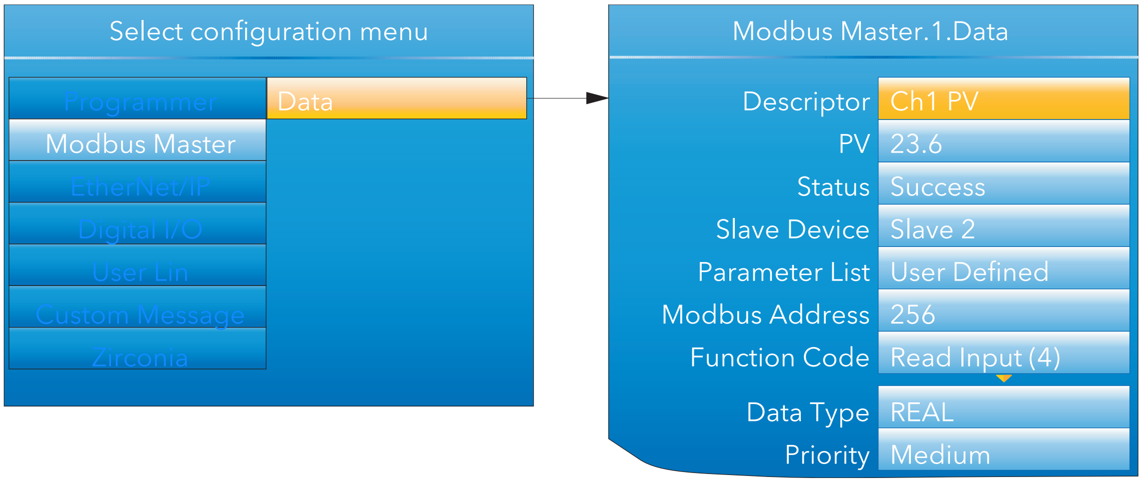



Figure 4.9.3b User defined parameters

DATA PARAMETERS

This lists all possible configuration fields that might appear, not just those shown in the examples above.

| | |
|---|---|
| Descriptor | Up to 20 characters used to describe the current data item (used in the Modbus Master user page (section 3.4.12)). |
| PV | The process value currently being read from the selected slave. Visible only if data item is not an alarm type. The value must be wired to a virtual channel with 'Operation' = 'Copy' if it is to be trended and/or recorded. |
| Sys Alm status | The status (e.g. None, Active) of the data item. Visible only for specific read profiles. The value must be wired to a virtual channel with 'Operation' = 'Copy' if it is to be trended and/or recorded. |
| Chan. Alm Status | The status of the data item. Visible only for specific read profiles. The value must be wired to a virtual channel with 'Operation' = 'Copy' if it is to be trended and/or recorded. |
| Set | Allows the user to set an on/off value. Visible only for specific write profiles. |
| Mode | Allows the user to set an auto/manual value. Visible only for specific write profiles. |
| Value | Configured or wired value to be sent to the selected slave. This parameter is available only with function codes 6 & 16 |
| Fall Back Value | The value to be sent to the selected slave if the 'Value' parameter is wired and has a status other than GOOD_PV. This parameter is available only with function codes 6 & 16 It is not possible to wire Fall Back Value from another parameter and it can be configured only manually |
| Send | A one shot action that sends the data in the 'Value' parameter or the 'Fall Back Value' parameter (depending upon the status of 'Value') to the selected slave. This is classed as an acyclic write and so is available only for function codes 6 & 16. The 'Priority' parameter must be set to 'Acyclic' |

### 4.9.3 MODBUS MASTER DATA CONFIGURATION (Cont.)

Status — The status of the last transaction sent to the selected slave

Success: The transaction was successfully actioned by the slave device

Timeout: There was no response from the slave device to a given request within the configured time

Illegal Address: The request to the slave device contained an invalid modbus address. The address may be for a read only parameter

Illegal Value: The request to the slave device contained invalid data for the specified parameter

Bad Sub: The sub function code in the request was invalid

Idle: This data item is currently idle and not communicating with the slave device
Illegal Code: The slave does not support the function code transmitted by the master.
Pending: The request is waiting to be sent, the most likely cause being that the slave device has not been set to online.

Slave Device — A list of available slaves that this data is to communicate with.

Parameter List — List of parameters available for the selected slave devices profile. These parameters require no user configuration.

Number — The channel, loop or group etc. instance.

Modbus Address — The Modbus register address that this data is to be read or written to. Limits are 0 - 65535

Function Code — The function code to use, this determines if the data is going to be read or written to the selected slave. Supported function codes are:

| Code | Description | Code | Description |
|---|---|---|---|
| 1 | Read contiguous status coils | 5 | Write a single coil on or off |
| 2 | Read contiguous discrete inputs | 6 | Write to a single register |
| 3 | Read contiguous holding registers | 8 | Loopback test |
| 4 | Read contiguous input registers | 16 | Write to contiguous registers |

Data Type — The data type that defines how this data is going to be represented. The data types listed below are supported.

8-bit signed byte (BYTE)

8-bit unsigned byte (UBYTE)

16-bit signed integer (INT)

16-bit unsigned integer (UINT)

32-bit signed long (DINT)

32-bit unsigned long (UDINT)

32-bit floating point IEEE (REAL)

32-bit signed long (little Endian, word swapped) (DINT (Swap))

32-bit unsigned long (little Endian, word swapped) (UDINT (Swap))

32-bit floating point IEEE (little Endian, word swapped) (REAL (swap))

Bit from register (BIT)

By default all 16 & 32 bit data types (unless specified) will be transmitted in Big Endian format, where the most significant byte in the value is sent first. Byte Ordering: (for big Endian) (0x12 sent first)

| | | |
|---|---|---|
| 16-bit | 0x1234 | 0x12, 0x34 |
| 32-bit | 0x12345678 | 0x12, 0x34, 0x56, 0x78 |

Bit Position — The bit in the register to be extracted, this is only available if the 'Data Type' selected is 'BIT In Register'

Scaling — The decimal placing for scaled 16 bit data types. Visible depending on the 'Data Type' selected.

Priority — The frequency with which this data will be managed. See 'Priority Levels', in section 4.9.1, above.

## 4.10 ETHERNET/IP CONFIGURATION

This area of configuration allows the 'Client' user to set up an EtherNet/IP communications link with up to two Server units. The 'Server' user has a more limited range of configurable items.

Note: Implicit I/O is used for continuous real-time transfer of multiple data items from instrument to instrument. Explicit I/O is used as a 'one-shot' transfer of a single data item. See section 3.4.13 for further details.

Figure 4.10 shows that the configuration is split into three areas: Main, Implicit Inputs and Implicit Outputs, but it should be noted that the implicit inputs and implicit outputs are read only, as these can be configured only by using iTools, as described in the EtherNet/IP display mode description (section 3.4.13).

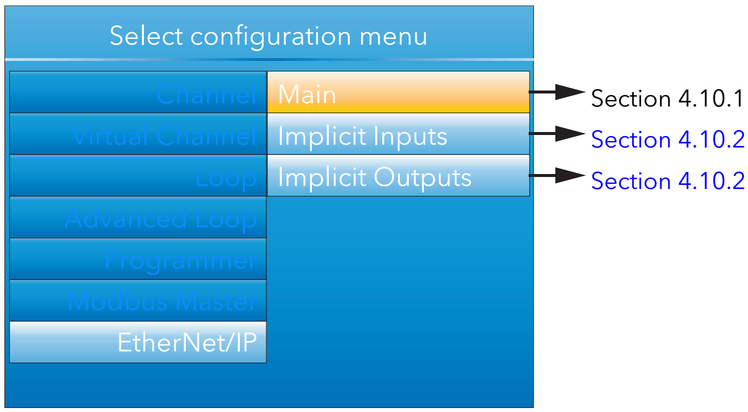



Figure 4.10 Client configuration

### 4.10.1 Ethernet/IP Configuration Main menu

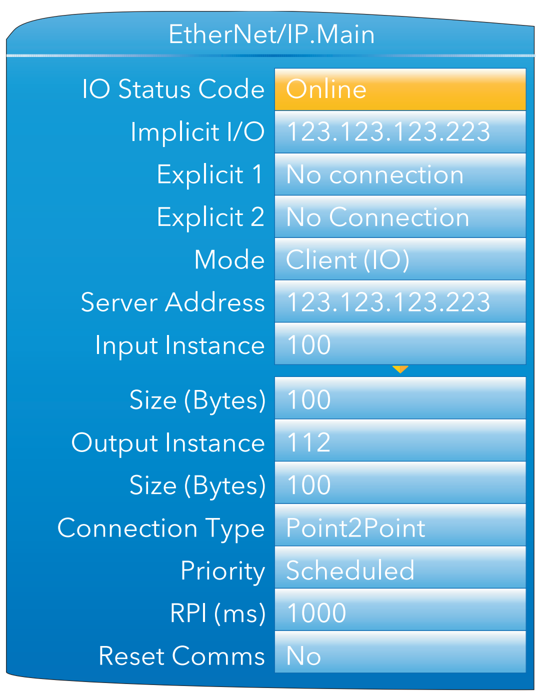



Figure 4.10.1 Ethernet/IP Main menu

### 4.10.1 ETHERNET/IP CONFIGURATION MAIN MENU (Cont.)

| | |
|---|---|
| Net Status Code | Network status (Server only)<br>Offline: nanodac online but there are currently no CIP connections<br>Online: nanodac online with at least 1 CIP connection<br>Connection Timeout: The connection has timed out<br>Duplicate IP: A duplicate IP address has been detected on the network<br>Initialisation: nanodac is initialising comms |
| IO Status Code | IO status (Client (IO) only). As above. |
| Tag Status code | Tag status (Client (Tags) only. See table 4.10.1, below. |
| Implicit I/O | Connected IO server IP address |
| Multicast | Connected IO server IP address (only if multicast selected) |
| Explicit 1 | Connected client/server IP address |
| Explicit 2 | Connected client/server IP address |
| Mode | Modes of operation: Server, Client (IO) or Client (Tags) |
| Server Address | IO Server IP address (Client mode only) |
| Input Instance | Input class instance number (client mode only) |
| Size (bytes) | The size in bytes of data that the client is expecting to read from the implicit input. |
| Output Instance | Output class instance number (client mode only) |
| Size (bytes) | The size of data that the client is expecting to write to the server. |
| Connection Type | Connection type (client mode only) |
| Priority | Connection priority (client mode only) |
| Rpi | IO connection speed (client mode only) |
| Reset Comms | Applies all changes to the EtherNet/IP stack at the same time. Or can be used to reset communications using the current configuration |
| Slot Number | PLC slot number (zero indexed) when communicating using tags |

### 4.10.2 Implicit inputs/outputs

This display provides a read-only display of the values in the input and output data tables. Parameters are placed in the input and output tables using the proprietary software tool called 'iTools', as described in section 3.4.13.

### 4.10.3 Explicit inputs/outputs

See section 3.4.13 for details.

### 4.10 ETHERNET/IP CONFIGURATION (Cont.)

| | |
|---|---|
| 0 | Success. Service was successful |
| 1 | Connection Failed. A connection in the path failed |
| 2 | Invalid Parameter. A parameter associated with the request was invalid |
| 3 | Memory Unavailable. No available resources in the server to service the request |
| 4 | Path Segment Error. The syntax of all or some of the path was not understood |
| 5 | Path Dest. Error. The path references an unknown object, class or instance |
| 6 | Partial Transfer. Only part of the expected data was transferred |
| 7 | Connection Lost. The messaging connection was lost |
| 8 | Service Unsupported. Undefined service for requested object |
| 9 | Invalid Attribute. Invalid attribute data detected |
| 10 | Attribute Error. An attribute in the response has a non zero status |
| 11 | Already Requested. The object is already in the mode/state being requested |
| 12 | Object Conflict. The object cannot perform the requested service |
| 13 | Already Exists. The requested instance or object already exists |
| 14 | Attribute Error. Request to modify a non modifiable attribute received |
| 15 | No Privileges. Permission/Privilege check failed |
| 16 | State Conflict. The current state or mode prohibits the execution of the requested service |
| 17 | Reply To Large. Response buffer too small for response data |
| 18 | Fragmented Value. For example this service request will return only half a REAL data type |
| 19 | Not Enough Data. The service does not provide enough data to complete the request |
| 20 | Invalid Attribute. Requested attribute is not supported |
| 21 | Too Much Data. The service supplied more than was expected |
| 22 | Object Non-Exist. The object specified does not exist in the device |
| 23 | Seq. Fragmentation. The fragmentation sequence for this service is not active |
| 24 | No Attribute Data. The attribute data for this object was not saved at the server prior to this request service |
| 25 | Data Store Failure. The attribute data for this object was not saved due to a failure during the attempt |
| 26 | Routing Failed. The service request packet was too large for transmission on a network in the path to the destination. The routing device was forced to abort the service |
| 27 | Routing Failed. The service request packet was too large for transmission on a network in the path to the destination. The routing device was forced to abort the service |
| 28 | Missing Attribute. The service did not supply an attribute in a list of attributes that was needed by the service to perform the requested behaviour |
| 29 | Invalid Attribute. The service is returning the list of attributes supplied with status information for those attributes that were invalid |
| 30 | Embedded Tag Error. An embedded service resulted in an error. This is most commonly an incorrectly formatted tag name |
| 31 | Vendor Error. A vendor specific error has encountered |
| 32 | Invalid Parameter. A parameter associated with the request was invalid |
| 33 | Write Once Error. An attempt to write to a write once only parameter occured |
| 34 | Invalid Reply. An invalid reply was received |
| 35 | Buffer Overflow. The message received is larger than the receiving buffer |
| 36 | Format Error. The format of the received message is not supported |
| 37 | Key Path Failure. The key segement in the path does not match destination key |
| 38 | Path Size Error. The size of the path in the request is too large |
| 39 | Unexpected Attribute. Unable to set the attribute at this time |
| 40 | Invalid Member Id. The requested member id does not match class object |
| 41 | Member Is R/O. A request to modify a R/O member was received |
| 42 | Group 2 Server. Group 2 DeviceNet server response |
| 43 | Translation Error. A CIP modbus translator request failed |
| 44 | Attribute Is R/O. A request to read a non readable attribute was received |
| 64 | No Tags Found. There were no tags configured in the input or output tables |
| 65 | Invalid Config. The total length in characters of all the tags in this table will cause the PLC to exceed its internal buffer of 500 bytes. To eliminate this problem, reduce the length of some or all tag names |

Table 4.10.1 Tag Status code definition

## 4.11 DIGITAL I/O

This area of configuration allows the digital I/O types to be selected.

Notes:
1. If 2A2B is set to 'Valve Raise', then 3A3B is set to 'Valve Lower'. Similarly, if relay 4AC is set to 'Valve Raise', then relay 5AC is set to 'Valve Lower'.
   When the loop channel output is wired to the PV input of a Valve Raise function, then the PV input of the associated Valve Lower function becomes unavailable for wiring, and both outputs are controlled by the loop as a pair, using only the single wire.
2. See section B2.6.11 for a description of time proportioning.

Note: These fields do not appear if they contain I/O type 'DC output'.

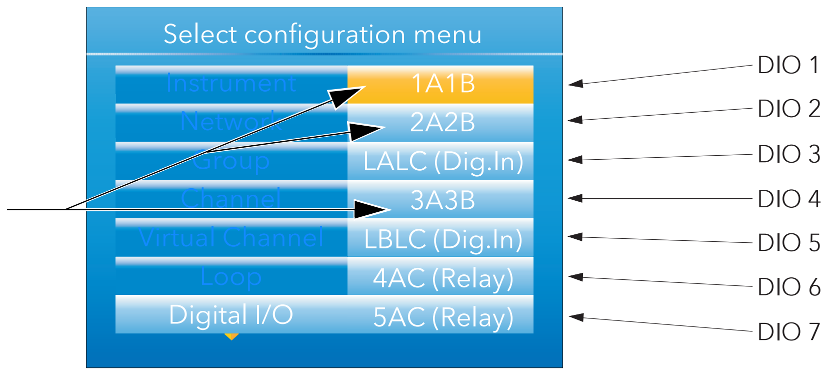



Figure 4.11 Digital I/O top level menu

### 4.11.1 Digital input/output

This applies to signals at terminals 1A/1B (figure 2.2). Highlight '1A1B', then operate the scroll key to reveal the configuration menu.

| | |
|---|---|
| Module Ident | Dig IO |
| Type | On Off O/P, Time Prop O/P or Contact I/P (default) |
| PV | For inputs, 0 = contact is open; 1 = contact is closed. For On Off O/P, a value ≥ 0.5 drives the output high, otherwise, the output is driven low. For Time Prop O/P, the value is the demanded output %. |
| Min On Time | For Type = Time Prop O/P only, this allows a minimum on time to be specified. Configurable range = 0.1 to 150 seconds |
| Invert | Inverts the output sense for digital outputs; or the input signal for digital inputs. |
| Output | Off = output being driven low; On = output being driven high. Does not appear for Type = Contact I/P |

### 4.11.2 Relay outputs

This may apply to terminal pairs 1A1B, 2A2B, 3A3B, 4AC, 5AC (figure 2.2). Highlight the relevant terminal pair, then operate the scroll key to reveal the configuration menu.

| | |
|---|---|
| Module Ident | Relay |
| Type (2A2B, 4AC) | On Off O/P (default), Time Prop O/P, Valve Raise (not if DC output I/O fitted). |
| Type (3A3B, 5AC) | 'On Off O/P' (default), 'Time Prop O/P'. The 3A3B relay is not fitted if 'DC Output' I/O is fitted. |
| PV | For On Off O/P, a value ≥ 0.5 closes the relay contacts, otherwise, the contacts are open. For Time Prop O/P, the value is the demanded output %. |
| Min On Time | For Type = Time Prop O/P only, this allows a minimum on time to be specified to reduce relay wear. Configurable range = 0.1 to 150 seconds |
| Invert | Inverts the output sense for the relays (not applicable if Type = Valve Raise). |

(Continued)

#### 4.11.2 RELAY OUTPUTS (Cont.)

| | |
|---|---|
| Inertia | For Type = Valve Raise only, this allows a value to be entered (in seconds) to take into account valve run-on. |
| Backlash | For Type = Valve Raise only, this allows a value to be entered (in seconds) in order to compensate for backlash in the valve linkage. |
| Standby action | For Type = Valve Raise only, this specifies the valve action when the instrument is in standby mode.<br>Continue: Output continues at the demanded level<br>Freeze: The valve stops being driven. |
| Output | Off = relay contacts open; On = relay contacts closed. |

### 4.11.3 Digital inputs

This applies to terminals pairs LALC, LBLC (figure 2.2). Highlight the relevant terminal pair, then operate the scroll key to reveal the configuration menu.

| | |
|---|---|
| Module Ident | Dig.In |
| Type | Contact I/P |
| PV | 0 = contact is open; 1 = contact is closed. |
| Invert | Inverts the sense of the input. |

### 4.11.4 Digital outputs

This applies to terminal pair 2A2B (figure 2.2). Highlight 2A2B, then operate the scroll key to reveal the configuration menu.

| | |
|---|---|
| Module Ident | Dig.Out |
| Type | On Off O/P, Time Prop O/P or Valve Raise |
| PV | For On Off O/P, a value $\geq 0.5$ drives the output high, otherwise, the output is driven low.<br>For Time Prop O/P, the value is the demanded output %. |
| Min On Time | For Type = Time Prop O/P only, this allows a minimum on time to be specified. Configurable range = 0.1 to 150 seconds |
| Invert | Inverts the output sense for digital outputs; or the input signal for digital inputs. |
| Inertia | For Type = Valve Raise only, this allows a value to be entered (in seconds) to take into account valve run-on. |
| Backlash | For Type = Valve Raise only, this allows a value to be entered (in seconds) in order to compensate for backlash in the valve linkage. |
| Standby action | For Type = Valve Raise only, this specifies the valve action when the instrument is in standby mode.<br>Continue: Output continues at the demanded level<br>Freeze: The valve stops being driven. |
| Output | Off = output being driven low; On = output being driven high. |

## 4.12 DC OUTPUT

This option provides a voltage (terminals 3A3B only) or mA output. Terminal location is shown in figure 2.2.

**CAUTION**

There are no mechanical interlocks to prevent a chassis with the dc output option being fitted into a 'sleeve' or 'case' which has previously been wired for the standard relay output. Before fitting the chassis into the case, it should be ensured that the terminal wiring is not attached to live voltage supplies, as such voltages may cause permanent damage to the instrument.

### 4.12.1 Configuration display

As shown in the figure below, highlight the required DC output, then operate the scroll button to reveal the configuration page.

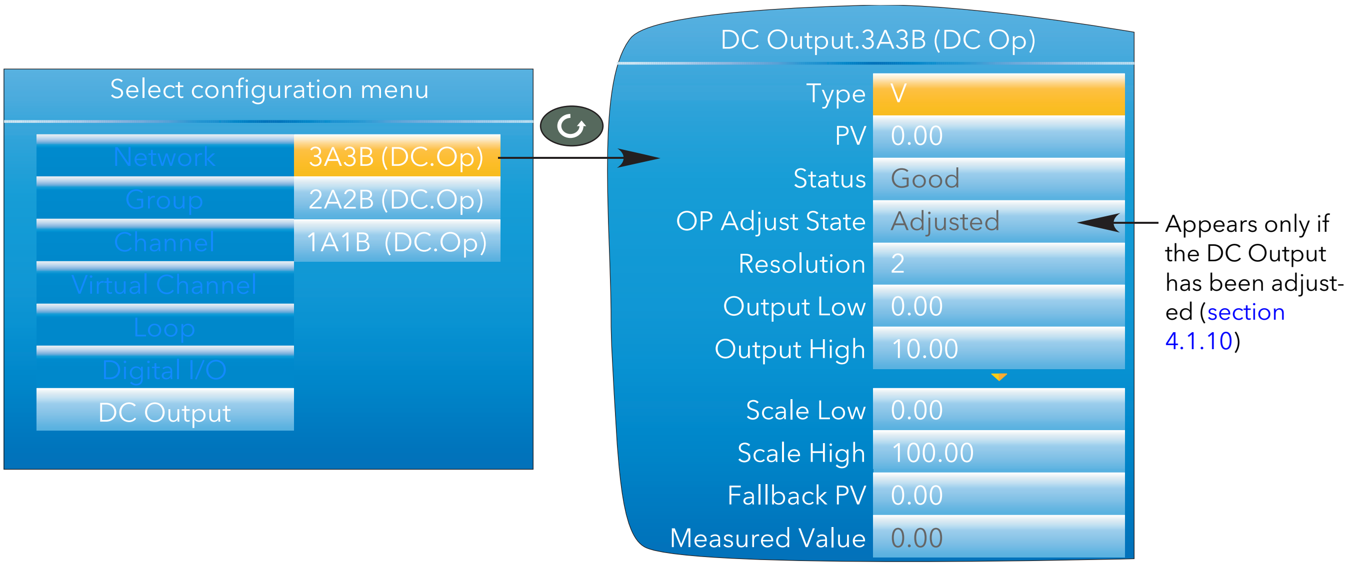



Figure 4.12.1 DC Output option configuration page (typical)

**PARAMETERS**

| | |
|---|---|
| Type | Select V(olts) (3A3B only) or mA as the output type. |
| PV | Input value to the function. Normally 'wired' to a suitable parameter. |
| Status | The status of the input parameter. |
| OP Adjust State | Adjusted. Appears only if the Output Adjust facility (section 4.1.10) has been used. |
| Resolution | The number of decimal places to be used for this configuration item. |
| Output Low | The minimum output value in Volts or mA as appropriate |
| Output High | The maximum output value to be output in Volts or mA as appropriate. |
| Scale Low | See 'SCALING INFORMATION' below. |
| Scale High | See 'SCALING INFORMATION' below. |
| Fallback PV | The output value when the status of the input parameter is not 'good'. |
| Measured Value | The Voltage or mA value appearing at the output terminals |

Note: The output voltage or current can be calibrated by using the output adjust procedure described in section 4.1.10.

**SCALING INFORMATION**

When PV = Scale Low, Output = output low value. When PV = Scale high, Output = output high value. The PV is mapped via the scale range onto the output range according to the equation:

$$\text{Output} = \left(\frac{\text{PV - Scale Low}}{\text{Scale High - Scale Low}}\right)(\text{Output High} - \text{Output Low}) + \text{Output Low}$$

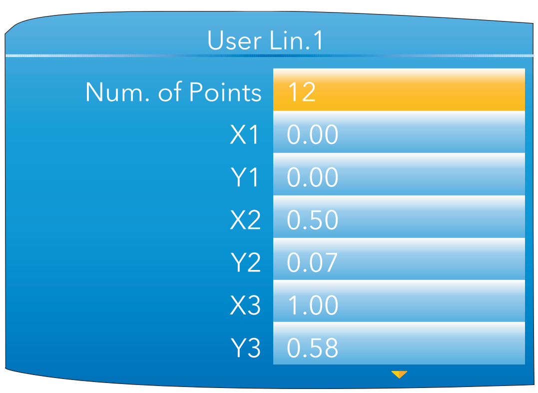
## 4.13 USER LIN

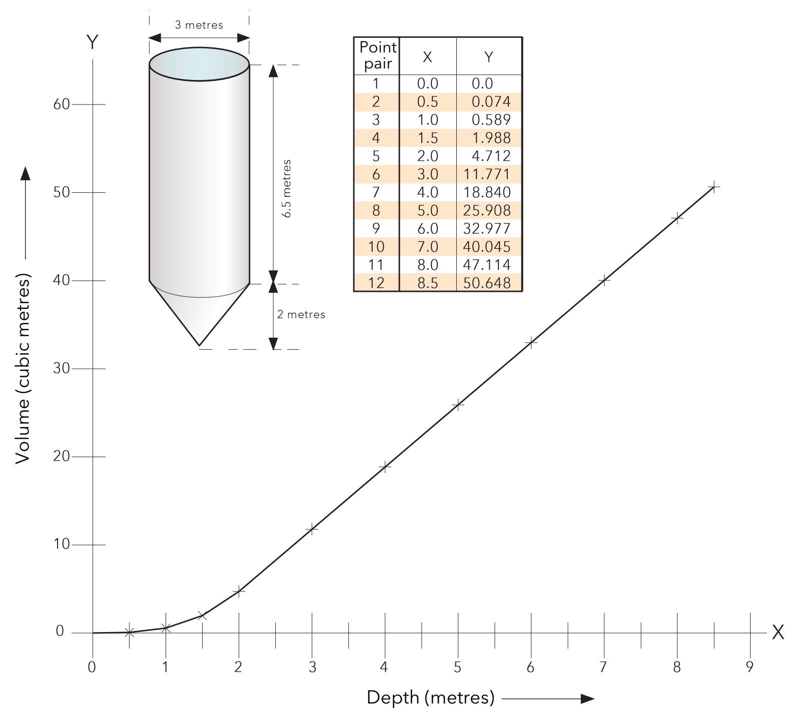
Allows the entry of up to four user linearisation tables, any one of which can be selected as 'Lin Type' in Channel configuration (section 4.4.1). Configuration consists of defining the number of points to be included (2 to 32) and then entering an X and a Y value for each point, where X values are the inputs and the Y values are the resulting outputs.

### 4.13.1 User linearisation table rules

1. Tables must be monotonic - i.e. there may not be more than one X value with the same Y value assigned to it.
2 Each X value must be greater than the preceding one.
3. Each Y value must be greater than the preceding one.
4. If units other than temperature units are to be displayed, the channel scale high and scale low values should be set to the same as the range high and low values, and the required scale units entered.

Figure 4.13.1 shows the first part of the configuration table for an imaginary cylinder example.




| Point pair | X | Y |
|---|---|---|
| 1 | 0.0 | 0.0 |
| 2 | 0.5 | 0.074 |
| 3 | 1.0 | 0.589 |
| 4 | 1.5 | 1.988 |
| 5 | 2.0 | 4.712 |
| 6 | 3.0 | 11.771 |
| 7 | 4.0 | 18.840 |
| 8 | 5.0 | 25.908 |
| 9 | 6.0 | 32.977 |
| 10 | 7.0 | 40.045 |
| 11 | 8.0 | 47.114 |
| 12 | 8.5 | 50.648 |

Figure 4.13.1 User Linearisation table example

When configuring a channel (section 4.4.1) to use a User linearisation table:

If Type = Thermocouple or RTD, then Range High/Low must be set to the highest and lowest 'Y' values to be used, respectively. The instrument automatically looks up the associated 'X' mV or Ohms values.

If Type = mV, V or mA, then Range High/Low must be set to the highest and lowest 'Y' values to be used, respectively. Input High/Low should be set to the highest and lowest 'X' values in the table, respectively.

## 4.14 CUSTOM MESSAGES

This feature allows the entry of up to 10 messages for sending to the history file, when triggered by a wired source (e.g. an alarm going active).

The messages of up to 100 characters each are entered using either the virtual keyboard, described in section 3.6, or by means of iTools configuration software.

Up to three parameter values may be embedded in messages in the format [*Address*], where 'Address' is the decimal Modbus address of the parameter (section5.3). E.G. [256] embeds Channel 1 PV.

## 4.15 ZIRCONIA BLOCK OPTION

This option allows the calculation of Carbon Potential, Dew point or Oxygen concentration. A zirconia (oxygen) probe consists of two platinum electrodes bonded to a pellet or cylinder of zirconia. At elevated temperatures, such a probe develops an emf across it which is proportional to the probe absolute temperature and to the log of the difference in oxygen partial pressure between its two ends.

The temperature of the probe is normally measured using a type K or type R thermocouple. The temperature effect on the thermocouple is such, that for successful operation, the probe temperature must be greater than 973K (700˚C).

### 4.15.1 Definitions

#### TEMPERATURE CONTROL

The sensor input of the temperature loop may come from the zirconia probe but it is common for a separate thermocouple to be used. The controller provides a heating output which may be used to control gas burners. In some applications a cooling output may also be connected to a circulation fan or exhaust damper.

#### CARBON POTENTIAL CONTROL

The zirconia probe generates a millivolt signal based on the ratio of oxygen concentrations on the reference side of the probe (outside the furnace) to the amount of oxygen in the furnace.

The controller uses the temperature and carbon potential signals to calculate the actual percentage of carbon in the furnace. This second loop generally has two outputs. One output is connected to a valve which controls the amount of an enrichment gas supplied to the furnace. The second output controls the level of dilution air.

#### SOOTING ALARM

In addition to other alarms which may be detected by the controller, the instrument can trigger an alarm when the atmospheric conditions are such that carbon will be deposited as soot on all surfaces inside the furnace. The alarm may be wired to an output (e.g. relay) to initiate an external alarm.

#### AUTOMATIC PROBE CLEANING

The instrument has a probe clean and recovery strategy that can be programmed to occur between batches or be manually requested. At the start of the cleaning process a ‘snapshot’ of the probe mV is taken, and a short blast of compressed air is used to remove any soot and other particles that may have accumulated on the probe. A minimum and maximum cleaning time can be set by the user. If the probe mV has not recovered to within 5% of the snapshot value within the maximum recovery time set then an alarm is given. This indicates that the probe is ageing and replacement or refurbishment is due. During the cleaning and recovery cycle the PV is frozen, thereby ensuring continuous furnace operation. The ‘Pv Frozen’ parameter can be used in an individual strategy, for example to hold the integral action during cleaning.

#### ENDOTHERMIC GAS CORRECTION

A gas analyser may be used to determine the carbon monoxide (CO) concentration of the endothermic gas. If a 4 to 20mA output is available from the analyser, this can be applied to the instrument to adjust the calculated % carbon reading automatically. Alternatively, this value can be entered manually.

#### CLEAN PROBE

As these sensors are used in furnace environments they require regular cleaning. Cleaning (Burn Off) is performed by forcing compressed air through the probe. Cleaning can be initiated either manually or automatically using a timed period. During cleaning ‘PV Frozen’ is set to ‘Yes’.

#### OXYGEN CONCENTRATION

In order to measure oxygen concentrations, one end of the probe is inserted into the atmosphere to be measured, whilst the other is subjected to a reference atmosphere. For most applications, air provides a suitable reference (reference input = 20.95 for air).

### 4.15.2 Configuration

The configuration parameters appear in one of three lists as shown in Figure 4.15.2a.

Zirconia

**Main**
- Probe type
- Resolution
- Min Calc Temp
- Tolerance
- Oxygen Exp
- Temp Input
- Probe Input
- Temp Offset
- Probe Offset
- Carbon Potential
- Dew Point
- Oxygen
- Soot Alarm
- PV Frozen
- Probe Status
- Balance Integral
- Carb Act CO O2
- Probe state
- Oxygen Type

**Gas Ref**
- CO Local
- CO Remote
- CO Remote Enable
- H2 Local
- H2 remote
- H2 Remote Enable
- H2 In use

**Clean**
- Clean Frequency
- Clean Time
- Min Rcv Time
- Max Rcv Time
- Clean Valve
- Clean Probe
- Time2Cln
- Clean Enable
- Clean Max Temp
- Clean Abort
- Clean Rcov Time
- Last Clean
- Clean Msg Reset
- Probe Fault
- Cant Clean
- Clean Abort
- Clean Temp

Figure 4.15.2a Zirconia probe configuration layout.

#### ZIRCONIA MAIN

The parameters that appear depend on the 'Probe Type' setting. For this reason, not all the parameters listed appear for all probe types. Figure 4.15.2b shows a typical configuration page.

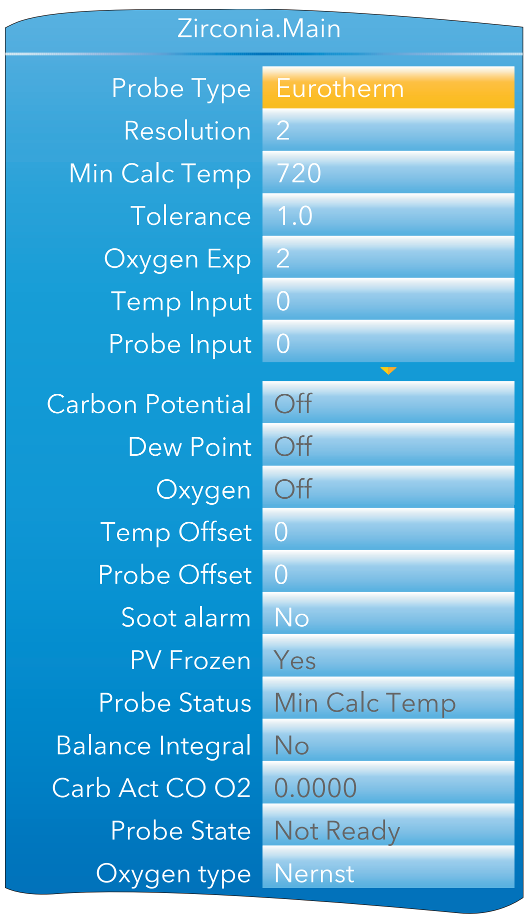



Figure 4.15.2b Zirconia Probe configuration (typical)

### 4.15.2 CONFIGURATION (Cont.)

#### MAIN PARAMETERS

| | |
|---|---|
| Probe Type | Select from a variety of probe manufacturers. The subsequent parameter list depends on which manufacturer is selected. |
| Resolution | Enter the number of decimal places to be used for the value display |
| Gas Reference | Reference value for the hydrogen concentration in the atmosphere. |
| Rem Gas Ref | Remote reference value for hydrogen concentration in the atmosphere. Allows hydrogen concentration to be read from an external source. |
| Rem Gas Enable | 'Yes' allows remote gas measurement. 'No' uses the internal Gas Reference value. |
| Working Gas | Read only. Working Reference Gas value |
| Min Calc Temp* | The minimum temperature in at which the calculation is valid. |
| Oxygen Exp | The exponent units of the log oxygen type calculation. valid entries -24 to +24. |
| Tolerance | Sooting tolerance multiplier. Allows the user to adjust the sensitivity of the Sooting alarm, in order to reduce the incidence of nuisance alarms. |
| Process Factor | Process factor defined by the probe manufacturer. |
| Clean Frequency | Allows the interval between probe cleaning cycles to be entered in hours and minutes. |
| Clean Time | Allows Probe clean time to be entered in hours and minutes. |
| Min Rcov Time | The minimum recovery time after purging in hours and minutes. |
| Max Rcov Time | The maximum recovery time after purging in hours and minutes. |
| Temp Input* | Zirconia probe temperature input value |
| Temp Offset* | Allows a temperature offset to be entered for the probe. |
| Probe Input | Zirconia probe mV input |
| Probe mV Offset | Allows an offset to be entered for the probe mV input |
| Oxygen | Read only. calculated oxygen value |
| Carbon Potential | Read only. The calculated carbon potential. |
| Dew Point | Read only. The dew point value derived from temperature and remote gas reference inputs. |
| Soot Alarm | Read only. Sooting alarm. Active if sooting is likely to take place. The sensitivity of the alarm can be adjusted by using the 'Tolerance' parameter, above. |
| Probe Fault | 'Yes' indicates a sensor break. |
| PV Frozen | Read only. Parameter set to 'Yes' during Probe cleaning. |
| Clean Valve | Read only. Enable the Clean valve. |
| Clean State | Read only. The burn off state of the zirconia probe: 'Waiting', 'Cleaning' or 'Recovering'.Clean Probe'Yes' = Initiate probe cleaning. 'No' = Do not clean probe. |
| Time to Clean | Read only. The time remaining, in hours and minutes until the next cleaning cycle is due. |
| Probe Status | Read only. Current probe status<br>OK — Normal working<br>mV Sensor Brk — Probe input sensor break<br>Temp Sensor Brk — Temperature input sensor break<br>Min Calc Temp — Probe deteriorating |
| Balance Integral | This output goes 'true' when a step change in the output occurs, which requires an integral re-balance if the readings are used for PID control. |
| Carb Act CO O2 | The carbon activity for the surface gas reaction between Carbon monoxide (CO) and Oxygen (O2) |
| Probe State | Read only. The current state of the probe. If 'Measuring', then the outputs are updated. For any other state (Clean, Clean Recovery, Test impedance, Impedance Recovery, Waiting), the outputs are not updated. |
| Oxygen Type | Oxygen equation being used. |

* Temperature units are those configured for the channel to which the temperature measuring transducer is connected.

### 4.15.2 CONFIGURATION (Cont.)

#### GAS REFERENCES PARAMETERS

| | |
|---|---|
| CO Local | Reference value for the carbon monoxide (CO) concentration in the atmosphere. |
| CO Remote | Remote reference value for the carbon monoxide concentration in the atmosphere. allows the value to be read remotely. |
| CO Remote En | 'Yes' allows remote CO measurement. 'No' uses the internal value. |
| CO in Use | The CO gas measurement value currently being used. |
| H2 Local | Reference value for the hydrogen (H) concentration in the atmosphere. |
| H2 Remote | Remote reference value for the hydrogen concentration in the atmosphere. allows the value to be read remotely. |
| H2 Remote En | 'Yes' allows remote H measurement. 'No' uses the internal value. |
| H2 In Use | The H gas measurement value currently being used. |

#### CLEAN PARAMETERS

| | |
|---|---|
| Clean Frequency | Allows the interval between probe cleaning cycles to be entered in hours and minutes. |
| Clean Time | Allows Probe clean time to be entered in hours and minutes. |
| Min Rcov Time | The minimum recovery time after purging in hours and minutes. |
| Max Rcov time | The maximum recovery time after purging in hours and minutes. |
| Clean Valve | Read only. Enable the Clean valve. |
| Clean Probe | Initiate probe cleaning |
| Time to Clean | Read only. The time remaining, in hours and minutes until the next cleaning cycle is due. |
| Clean Enable | Enable probe cleaning |
| Clean Max Temp | Maximum temperature for cleaning. If the temperature exceeds this value, cleaning is aborted. |
| Clean Abort | Abort probe cleaning |
| Clean Rcov Time | The time taken for the probe to recover to 95% of its original value after the last clean. If the last clean did not recover within the Max Rcov time, this value is set to 0. |
| Last Clean | The mV output from the probe after the last clean. |
| Clean Msg Reset | 'Yes' clears cleaning related alarms |
| Probe Fault | 'Yes' means that the probe failed to recover to 95% of its original output, following a clean, |
| Cant Clean | Conditions exist which prevent a clean cycle starting. Can be cleared using 'Clean Msg Reset'. |
| Clean Abort | A clean cycle was aborted. Can be cleared using 'Clean Msg Reset'. |
| Clean Temp | A clean cycle was aborted because the temperature was too high. Can be cleared using 'Clean Msg Reset'. |

### 4.15.3 Wiring

Figure 4.15.3 shows a typical wiring arrangement for a Zirconia probe.

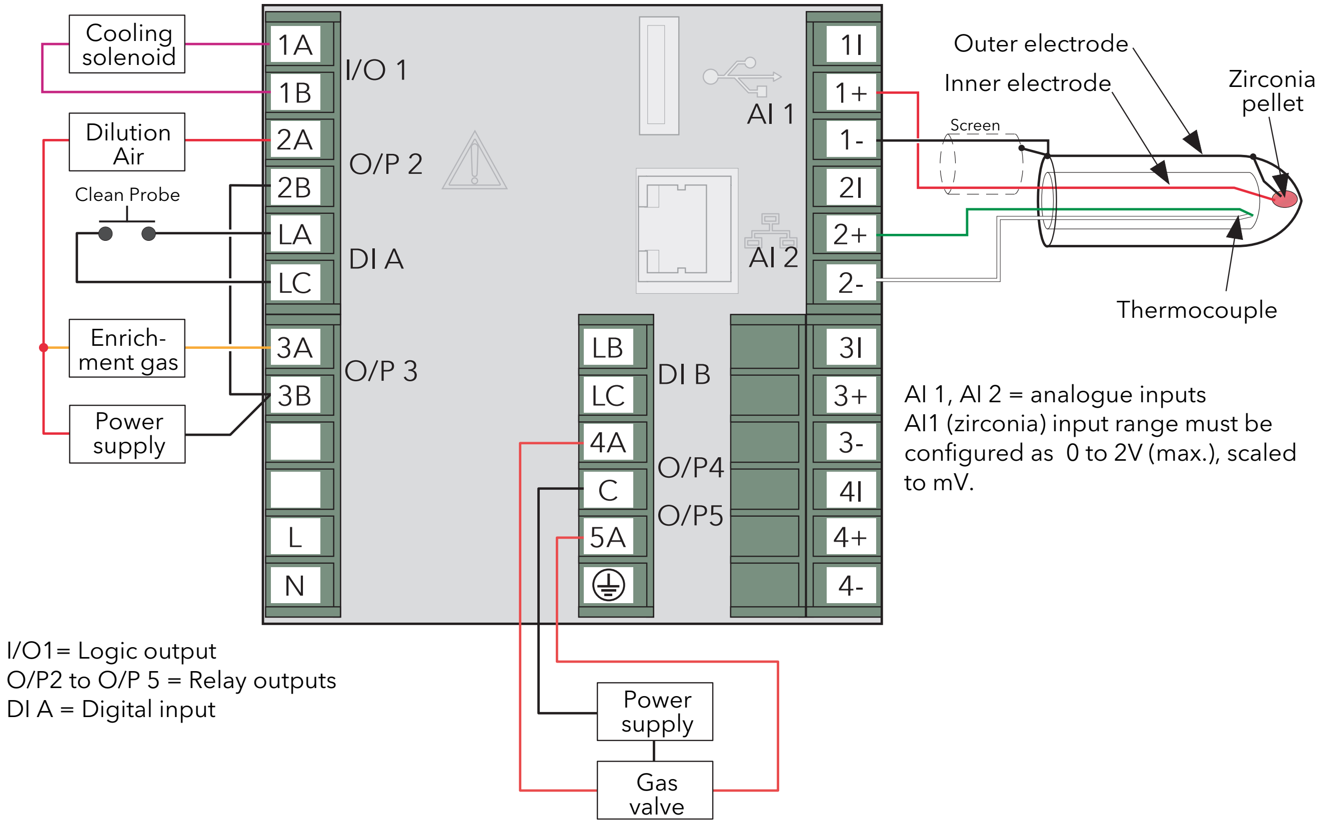



Figure 4.15.3 Typical zirconia probe wiring

## 4.16 STERILISER OPTION

This block provides a means of recording complete sterilisation cycles, including for example, venting and pumping as well as the actual sterilising period. See section 3.4.10 for display mode details.

Data is stored in .uhh history files for viewing in Review software.

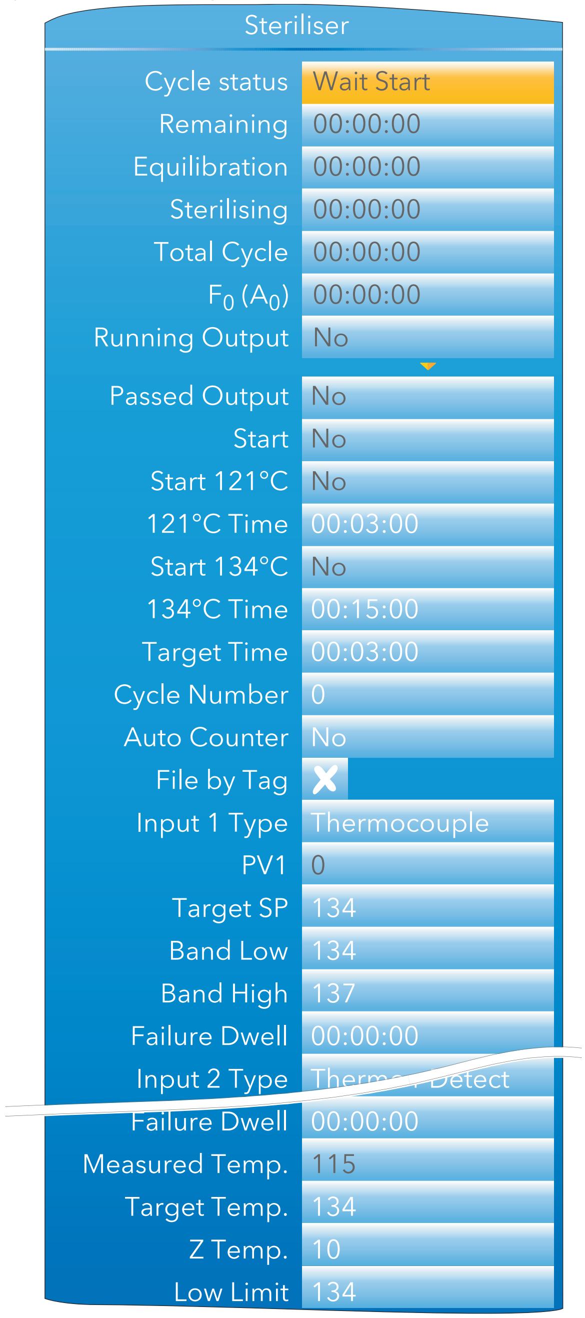



Figure 4.16 Steriliser block configuration menu

### 4.16.1 Configuration parameters

| | |
|---|---|
| Cycle Status | Wait start: The cycle is waiting to be started |
| | Waiting: Waiting for input 1 to reach its target setpoint. |
| | Equilibration: Currently in the equilibration period |
| | Sterilising: Currently in the sterilising phase |
| | Passed: The cycle has completed successfully |
| | Failed: The cycle has failed |
| | Test cycle: A test cycle is in progress |

### 4.16.1 CONFIGURATION PARAMETERS (Cont.)

| | |
|---|---|
| Remaining | The sterilising time remaining for the current cycle |
| Equilibration | The equilibration time period for the current cycle |
| Sterilising | The time for which the load has currently been at sterilisation conditions |
| Total Cycle | The total cycle time |
| $F_0$ ($A_0$) | The current $F_0$, $F_H$ or $A_0$ value |
| Running Output | 'Yes' = Cycle running; 'No' = Cycle not running |
| Passed Output | 'Yes' = Output passed; 'No' = Output did not pass |
| Start | Trigger to start a custom cycle (i.e. one for which High and Low band and / or Target setpoint have been changed from their default values.) |
| Start 121°C | Trigger to start a pre-defined 121°C cycle (Setpoint, Band Low/Band High etc. values are set to their 121° defaults when the cycle is initiated). |
| 121°C Time | Target time for a121°C cycle. Automatically copied to the 'Target Time' field when Start 121°C requested. Scrollable value in hh:mm:ss format. |
| Start 134°C | Trigger to start a pre-defined 134°C cycle (Setpoint, Band Low/Band High etc. values are set to their 134° defaults when the cycle is initiated) |
| 134°C Time | Target time for a134°C cycle. Automatically copied to the 'Target Time' field when Start 134°C requested. Scrollable value in hh:mm:ss format. |
| Target Time | The time for which the input values must remain at their sterilisation values in order that the cycle shall pass. The cycle fails if any input moves outside its specified band limits during the Target Time. Scrollable value in hh:mm:ss format. |
| Cycle Number | Each execution of the Steriliser block uses a unique cycle number. This may be entered manually, or can be set to increment automatically by setting 'Auto Counter' (below) to 'Yes'. |
| Auto Counter | 'Yes' causes the Cycle Number (above) to increment automatically each time a new cycle is initiated. If Auto counter = 'Yes', the Cycle Number forms part of the historical data and can be used to help identify data during later review. |
| File By Tag | 'Tick' ensures that each cycle is recorded in its own unique history file identified by cycle number and 'File tag' (below). |
| File tag | This field appears only if 'File By Tag' is enabled (tick symbol). File tag allows a four-character identifier to be entered to be used with the Cycle Number (above) to identify the history file |
| Input n Type | Select 'Off', 'Thermocouple', 'Rising Pressure', 'Falling pressure', 'Rise Air Detect', or 'Fall Air Detect'.<br>Off: This input will not be included in steriliser monitoring calculations<br>Thermocouple: Degrees Celsius input<br>Rising pressure: A mBar pressure input with a rising pressure expected during the cycle. This pressure input would normally be synchronised with a temperature input, in the same chamber, when performing a 121°C or 134°C cycle.<br>Falling pressure: As 'Rising Pressure' above, but with a falling pressure expected during the cycle<br>Rise Air Detect: A mBar pressure input with a rising pressure expected during the cycle. This pressure input is not synchronised with a temperature input when performing a 121°C or 134°C cycle, as it is (typically) an outside chamber pressure.<br>Fall Air Detect: As 'Rise Air Detect' above, but with a falling pressure expected during the cycle |
| PV n | Input value (wireable only). See note 1 below. |
| Target SP | Target setpoint for this input. (Does not appear if relevant Input Type = 'Off'.) See note 2 below. |
| Band Low/High | The low and high steriliser temperature or pressure band for this input. (Does not appear if relevant Input Type = 'Off'.) See note 2 below. Values are effective only during Sterilisation mode. |

### 4.16.1 CONFIGURATION PARAMETERS (Cont.)

| | |
|---|---|
| Failure Dwell | A failure alarm is set if this input is out of band range for more than the Failure Dwell time. Scrollable value in hh:mm:ss format. |

Notes
1. n = 1 to 4, where typically, inputs 1 to 3 are temperature inputs and input 4 is a pressure input.
2. Target SP and Band High/Low values are set to their relevant default values when a 121°C or 134°C cycle is initiated.

| | |
|---|---|
| Measured Temp. | For $F_0$ or $A_0$ calculations, this value must be in °C. Typically wired to an input channel PV . |
| Target Temp. | For $F_0$ or $A_0$ calculations, the target temperature (see section 3.4.10 for details). This typically is the same value as the Target SP (above). |
| Z Temp. | For $F_0$ or $A_0$ calculations this is a temperature interval representing a factor-of-10 increase in killing efficiency. Z = 10°C for $F_0$ and $A_0$, and 20°C for $F_H$ |
| Low Limit | The temperature below which $F_0$ or $A_0$ calculations are suspended. |

## 4.17 HUMIDITY BLOCK OPTION

This block uses wet and dry bulb temperatures, and atmospheric pressure inputs to derive values for relative humidity and dew point.

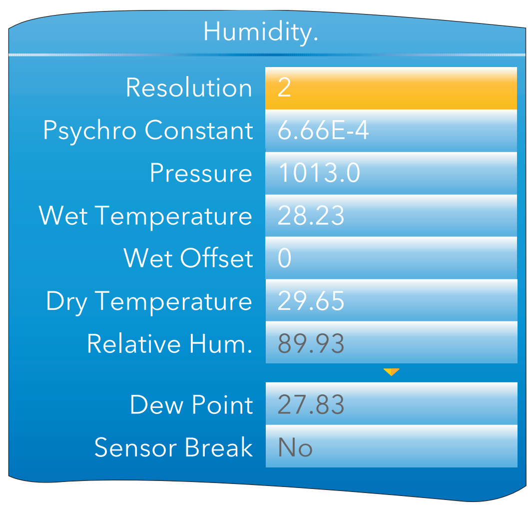



Figure 4.17 Humidity calculation configuration

### 4.17.1 Configuration parameters

| | |
|---|---|
| Resolution | The number of decimal places for the Relative humidity and Dew point displays. |
| Psychro constant | The psychrometric constant (default = 6.66 x $10^{-4}$) (See note below). |
| Pressure | The current atmospheric pressure in mBar. |
| Wet Temperature | The wet bulb thermometer temperature. |
| Wet Offset | Offset for the wet bulb temperature. |
| Dry Temperature | The dry bulb thermometer temperature. |
| Relative Hum. | The relative humidity value calculated from the Wet temperature, the Dry temperature and the Pressure inputs. The number of decimal places depends on the Resolution setting. |
| Dew Point | The dew point value calculated from the Wet temperature, the Dry temperature and the Pressure inputs. The number of decimal places depends on the Resolution setting. |
| Sensor Break | 'Yes' implies that a break has occurred between one (or more) of the temperature or pressure transducer and its input. |

Note: The default value 6.66 may be edited, but the multiplier is always $10^{-4}$ (i.e. it cannot be edited).

## 4.18 BCD INPUT

Part of the 'Toolkit Blocks' option, this block derives decimal and two-decade binary coded decimal (BCD) values from eight discrete inputs, where input 1 is the least significant input ($2^0$ = 1) and input 8 is the most significant ($2^7$ = 128). The example below shows that for inputs 2, 4, 6 and 8 high, the decimal input value is 170, but the BCD value is invalid. In any such case, the maximum BCD value for each decade is limited to 9.

| Input number | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | |
|---|---|---|---|---|---|---|---|---|---|
| Input status | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | |
| Decimal input | 128 | 0 | 32 | 0 | 8 | 0 | 2 | 0 | (=170) |
| BCD output | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | (=10, 10) |

Figure 4.18 BCD block example

### 4.18.1 Input rules

Valid BCD outputs are produced only with the following inputs set:

1. Any combination of inputs 1, 2, 3, 5, 6 and 7
2. Any combination of Inputs 1, 4, 5 and 8

### 4.18.2 Configuration

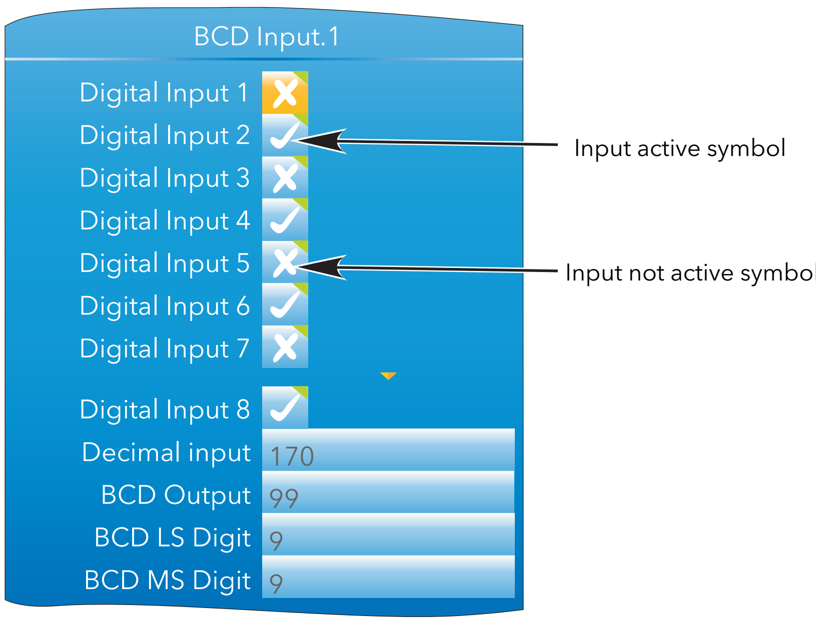



Figure 4.18.2 BCD block configuration

**PARAMETERS**

| | |
|---|---|
| Digital Input N | Digital inputs, wired (for example) to contact inputs at the rear panel or to other suitable parameter outputs. |
| Decimal input | The value defined by the active inputs, where input 1 = 1, when active, input 2 = 2, input 3 = 4, input 4 = 8 and so on. |
| BCD Output | A two digit output being the binary coded decimal version of the input. |
| BCD LS Digit | This least significant (right-most) digit represents the value of inputs 1 to 4, where input 1 = 1, input 2 = 2, input 3 = 4, input 4 = 8. Maximum value = 9, even if input is greater than 9. |
| BCD MS Digit | This most significant (left-most) digit represents the value of inputs 5 to 8, where input 5 = 1, input 6 = 2, input 7 = 4, input 8 = 8. Maximum value = 9, even if input is greater than 9. |

## 4.19 LOGIC (2 INPUT) BLOCK

Part of the 'Toolkit Blocks' option, this block allows a number of logic and comparison operations to be performed on a pair of inputs. For logic functions, the inputs can be inverted to allow, for example, a NOR function to be implemented by inverting the inputs to an AND function. 12 two-input logic blocks are available.

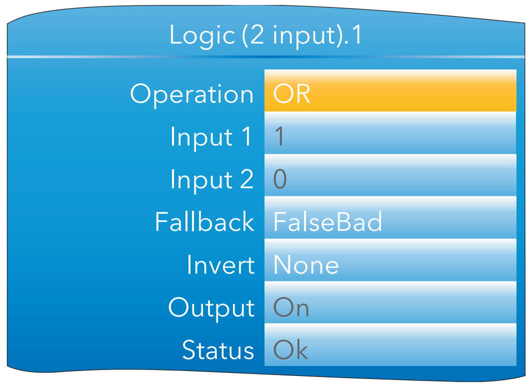



Figure 4.19 Two-input logic block configuration

### 4.19.1 Parameters

| | |
|---|---|
| Operation | AND, OR, XOR, LATCH (boolean values only)<br>== (Input 1 = Input 2)<br><> (Input 1 ≠ Input 2)<br>< (Input 1 < Input 2}<br><= (Input 1 ≤ Input 2)<br>> (Input 1 > Input 2)<br>=> (Input 1 ≥ Input 2) |
| Input 1(2) | The inputs to the specified operation. For inverted inputs (below), this shows the 'real' (non-inverted) state. |
| Fallback | Configures the output and status values to be used if either input has a status other than 'Good'.<br>FalseBad: Output = False; Status = Bad<br>TrueBad: Output = True; Status = Bad<br>FalseGood: Output = False; Status = Good<br>TrueGood: Output = True; Status = Good |
| Invert | For logic operators only allows neither, either or both inputs to be inverted. Input 1 and Input 2 show the non-inverted state. |
| Output | On or Off depending on input states etc. |
| Status | The status of the result ('Ok' or 'Error'). |

## 4.20 LOGIC (8 INPUT) BLOCK

Part of the 'Toolkit Blocks' option, this block allows AND, OR and cascading* XOR logic operations to be carried out on up to eight inputs.

*Cascading XOR example for inputs 1 to 4: (((Input1 ⊕ Input2) ⊕ Input3) ⊕ Input4).

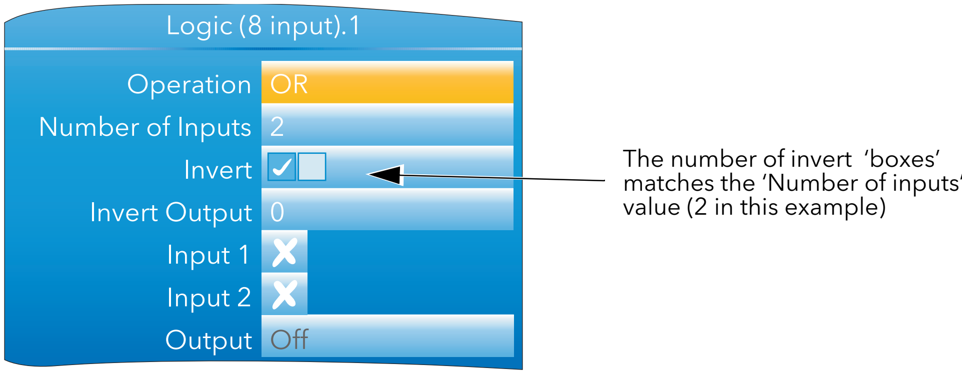



Figure 4.20 Eight input logic block configuration

### 4.20.1 Parameters

| | |
|---|---|
| Operation | AND, OR or XOR |
| Number of inputs | The number of inputs to the logic operator |
| Invert | Allows the user to invert individual inputs, as described below. |
| Invert Output | 'Yes' inverts the output status |
| Input 1 | The status of input 1, ignoring the Invert status. Cross = off; Tick = on. |
| Inputs 2 to N | As for input 1, where N = the value of the 'Number of Inputs' parameter. |
| Output | On or Off. Includes the effect of 'Invert Output' status. |

#### INPUT INVERSION

1. Use the down arrow key to highlight the 'Invert' field and operate the scroll key to enter edit mode
2. Use the up arrow key to highlight the first input to be inverted (the relevant input numbers appear in the display boxes for uninverted inputs when highlighted).
3. Once the required input box is highlighted, use the scroll key to change the numeric character to a tick symbol (to invert) or change the tick character to a numeric character (to remove a previous inversion).
4. Repeat for any further inputs, then operate the page key to confirm the changes and to quit edit mode.

### 4.20.2 Schematic

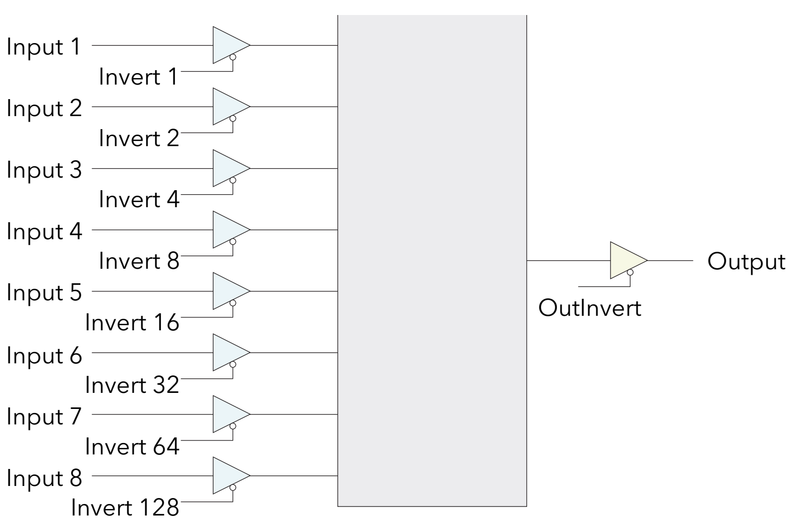



Figure 4.20.2 Logic (8 input) block schematic

### 4.20.3 Invert input decoding table

Over a communications link, the inversion status is transmitted as a decimal value, which can be encoded/decoded using the following table

| Input<br>8 7 6 5 4 3 2 1 | Hex | Dec | Input<br>8 7 6 5 4 3 2 1 | Hex | Dec | Input<br>8 7 6 5 4 3 2 1 | Hex | Dec | Input<br>8 7 6 5 4 3 2 1 | Hex | Dec |
|---|---|---|---|---|---|---|---|---|---|---|---|
| N N N N N N N N | 00 | 0 | N 7 N N N N N N | 40 | 64 | 8 N N N N N N N | 80 | 128 | 8 7 N N N N N N | C0 | 192 |
| N N N N N N N 1 | 01 | 1 | N 7 N N N N N 1 | 41 | 65 | 8 N N N N N N 1 | 81 | 129 | 8 7 N N N N N 1 | C1 | 193 |
| N N N N N N 2 N | 02 | 2 | N 7 N N N N 2 N | 42 | 66 | 8 N N N N N 2 N | 82 | 130 | 8 7 N N N N 2 N | C2 | 194 |
| N N N N N N 2 1 | 03 | 3 | N 7 N N N N 2 1 | 43 | 67 | 8 N N N N N 2 1 | 83 | 131 | 8 7 N N N N 2 1 | C3 | 195 |
| N N N N N 3 N N | 04 | 4 | N 7 N N N 3 N N | 44 | 68 | 8 N N N N 3 N N | 84 | 132 | 8 7 N N N 3 N N | C4 | 196 |
| N N N N N 3 N 1 | 05 | 5 | N 7 N N N 3 N 1 | 45 | 69 | 8 N N N N 3 N 1 | 85 | 133 | 8 7 N N N 3 N 1 | C5 | 197 |
| N N N N N 3 2 N | 06 | 6 | N 7 N N N 3 2 N | 46 | 70 | 8 N N N N 3 2 N | 86 | 134 | 8 7 N N N 3 2 N | C6 | 198 |
| N N N N N 3 2 1 | 07 | 7 | N 7 N N N 3 2 1 | 47 | 71 | 8 N N N N 3 2 1 | 87 | 135 | 8 7 N N N 3 2 1 | C7 | 199 |
| N N N N 4 N N N | 08 | 8 | N 7 N N 4 N N N | 48 | 72 | 8 N N N 4 N N N | 88 | 136 | 8 7 N N 4 N N N | C8 | 200 |
| N N N N 4 N N 1 | 09 | 9 | N 7 N N 4 N N 1 | 49 | 73 | 8 N N N 4 N N 1 | 89 | 137 | 8 7 N N 4 N N 1 | C9 | 201 |
| N N N N 4 N 2 N | 0A | 10 | N 7 N N 4 N 2 N | 4A | 74 | 8 N N N 4 N 2 N | 8A | 138 | 8 7 N N 4 N 2 N | CA | 202 |
| N N N N 4 N 2 1 | 0B | 11 | N 7 N N 4 N 2 1 | 4B | 75 | 8 N N N 4 N 2 1 | 8B | 139 | 8 7 N N 4 N 2 1 | CB | 203 |
| N N N N 4 3 N N | 0C | 12 | N 7 N N 4 3 N N | 4C | 76 | 8 N N N 4 3 N N | 8C | 140 | 8 7 N N 4 3 N N | CC | 204 |
| N N N N 4 3 N 1 | 0D | 13 | N 7 N N 4 3 N 1 | 4D | 77 | 8 N N N 4 3 N 1 | 8D | 141 | 8 7 N N 4 3 N 1 | CD | 205 |
| N N N N 4 3 2 N | 0E | 14 | N 7 N N 4 3 2 N | 4E | 78 | 8 N N N 4 3 2 N | 8E | 142 | 8 7 N N 4 3 2 N | CE | 206 |
| N N N N 4 3 2 1 | 0F | 15 | N 7 N N 4 3 2 1 | 4F | 79 | 8 N N N 4 3 2 1 | 8F | 143 | 8 7 N N 4 3 2 1 | CF | 207 |
| N N N 5 N N N N | 10 | 16 | N 7 N 5 N N N N | 50 | 80 | 8 N N 5 N N N N | 90 | 144 | 8 7 N 5 N N N N | D0 | 208 |
| N N N 5 N N N 1 | 11 | 17 | N 7 N 5 N N N 1 | 51 | 81 | 8 N N 5 N N N 1 | 91 | 145 | 8 7 N 5 N N N 1 | D1 | 209 |
| N N N 5 N N 2 N | 12 | 18 | N 7 N 5 N N 2 N | 52 | 82 | 8 N N 5 N N 2 N | 92 | 146 | 8 7 N 5 N N 2 N | D2 | 210 |
| N N N 5 N N 2 1 | 13 | 19 | N 7 N 5 N N 2 1 | 53 | 83 | 8 N N 5 N N 2 1 | 93 | 147 | 8 7 N 5 N N 2 1 | D3 | 211 |
| N N N 5 N 3 N N | 14 | 20 | N 7 N 5 N 3 N N | 54 | 84 | 8 N N 5 N 3 N N | 94 | 148 | 8 7 N 5 N 3 N N | D4 | 212 |
| N N N 5 N 3 N 1 | 15 | 21 | N 7 N 5 N 3 N 1 | 55 | 85 | 8 N N 5 N 3 N 1 | 95 | 149 | 8 7 N 5 N 3 N 1 | D5 | 213 |
| N N N 5 N 3 2 N | 16 | 22 | N 7 N 5 N 3 2 N | 56 | 86 | 8 N N 5 N 3 2 N | 96 | 150 | 8 7 N 5 N 3 2 N | D6 | 214 |
| N N N 5 N 3 2 1 | 17 | 23 | N 7 N 5 N 3 2 1 | 57 | 87 | 8 N N 5 N 3 2 1 | 97 | 151 | 8 7 N 5 N 3 2 1 | D7 | 215 |
| N N N 5 4 N N N | 18 | 24 | N 7 N 5 4 N N N | 58 | 88 | 8 N N 5 4 N N N | 98 | 152 | 8 7 N 5 4 N N N | D8 | 216 |
| N N N 5 4 N N 1 | 19 | 25 | N 7 N 5 4 N N 1 | 59 | 89 | 8 N N 5 4 N N 1 | 99 | 153 | 8 7 N 5 4 N N 1 | D9 | 217 |
| N N N 5 4 N 2 N | 1A | 26 | N 7 N 5 4 N 2 N | 5A | 90 | 8 N N 5 4 N 2 N | 9A | 154 | 8 7 N 5 4 N 2 N | DA | 218 |
| N N N 5 4 N 2 1 | 1B | 27 | N 7 N 5 4 N 2 1 | 5B | 91 | 8 N N 5 4 N 2 1 | 9B | 155 | 8 7 N 5 4 N 2 1 | DB | 219 |
| N N N 5 4 3 N N | 1C | 28 | N 7 N 5 4 3 N N | 5C | 92 | 8 N N 5 4 3 N N | 9C | 156 | 8 7 N 5 4 3 N N | DC | 220 |
| N N N 5 4 3 N 1 | 1D | 29 | N 7 N 5 4 3 N 1 | 5D | 93 | 8 N N 5 4 3 N 1 | 9D | 157 | 8 7 N 5 4 3 N 1 | DD | 221 |
| N N N 5 4 3 2 N | 1E | 30 | N 7 N 5 4 3 2 N | 5E | 94 | 8 N N 5 4 3 2 N | 9E | 158 | 8 7 N 5 4 3 2 N | DE | 222 |
| N N N 5 4 3 2 1 | 1F | 31 | N 7 N 5 4 3 2 1 | 5F | 95 | 8 N N 5 4 3 2 1 | 9F | 159 | 8 7 N 5 4 3 2 1 | DF | 223 |
| N N 6 N N N N N | 20 | 32 | N 7 6 N N N N N | 60 | 96 | 8 N 6 N N N N N | A0 | 160 | 8 7 6 N N N N N | E0 | 224 |
| N N 6 N N N N 1 | 21 | 33 | N 7 6 N N N N 1 | 61 | 97 | 8 N 6 N N N N 1 | A1 | 161 | 8 7 6 N N N N 1 | E1 | 225 |
| N N 6 N N N 2 N | 22 | 34 | N 7 6 N N N 2 N | 62 | 98 | 8 N 6 N N N 2 N | A2 | 162 | 8 7 6 N N N 2 N | E2 | 226 |
| N N 6 N N N 2 1 | 23 | 35 | N 7 6 N N N 2 1 | 63 | 99 | 8 N 6 N N N 2 1 | A3 | 163 | 8 7 6 N N N 2 1 | E3 | 227 |
| N N 6 N N 3 N N | 24 | 36 | N 7 6 N N 3 N N | 64 | 100 | 8 N 6 N N 3 N N | A4 | 164 | 8 7 6 N N 3 N N | E4 | 228 |
| N N 6 N N 3 N 1 | 25 | 37 | N 7 6 N N 3 N 1 | 65 | 101 | 8 N 6 N N 3 N 1 | A5 | 165 | 8 7 6 N N 3 N 1 | E5 | 229 |
| N N 6 N N 3 2 N | 26 | 38 | N 7 6 N N 3 2 N | 66 | 102 | 8 N 6 N N 3 2 N | A6 | 166 | 8 7 6 N N 3 2 N | E6 | 230 |
| N N 6 N N 3 2 1 | 27 | 39 | N 7 6 N N 3 2 1 | 67 | 103 | 8 N 6 N N 3 2 1 | A7 | 167 | 8 7 6 N N 3 2 1 | E7 | 231 |
| N N 6 N 4 N N N | 28 | 40 | N 7 6 N 4 N N N | 68 | 104 | 8 N 6 N 4 N N N | A8 | 168 | 8 7 6 N 4 N N N | E8 | 232 |
| N N 6 N 4 N N 1 | 29 | 41 | N 7 6 N 4 N N 1 | 69 | 105 | 8 N 6 N 4 N N 1 | A9 | 169 | 8 7 6 N 4 N N 1 | E9 | 233 |
| N N 6 N 4 N 2 N | 2A | 42 | N 7 6 N 4 N 2 N | 6A | 106 | 8 N 6 N 4 N 2 N | AA | 170 | 8 7 6 N 4 N 2 N | EA | 234 |
| N N 6 N 4 N 2 1 | 2B | 43 | N 7 6 N 4 N 2 1 | 6B | 107 | 8 N 6 N 4 N 2 1 | AB | 171 | 8 7 6 N 4 N 2 1 | EB | 235 |
| N N 6 N 4 3 N N | 2C | 44 | N 7 6 N 4 3 N N | 6C | 108 | 8 N 6 N 4 3 N N | AC | 172 | 8 7 6 N 4 3 N N | EC | 236 |
| N N 6 N 4 3 N 1 | 2D | 45 | N 7 6 N 4 3 N 1 | 6D | 109 | 8 N 6 N 4 3 N 1 | AD | 173 | 8 7 6 N 4 3 N 1 | ED | 237 |
| N N 6 N 4 3 2 N | 2E | 46 | N 7 6 N 4 3 2 N | 6E | 110 | 8 N 6 N 4 3 2 N | AE | 174 | 8 7 6 N 4 3 2 N | EE | 238 |
| N N 6 N 4 3 2 1 | 2F | 47 | N 7 6 N 4 3 2 1 | 6F | 111 | 8 N 6 N 4 3 2 1 | AF | 175 | 8 7 6 N 4 3 2 1 | EF | 239 |
| N N 6 5 N N N N | 30 | 48 | N 7 6 5 N N N N | 70 | 112 | 8 N 6 5 N N N N | B0 | 176 | 8 7 6 5 N N N N | F0 | 240 |
| N N 6 5 N N N 1 | 31 | 49 | N 7 6 5 N N N 1 | 71 | 113 | 8 N 6 5 N N N 1 | B1 | 177 | 8 7 6 5 N N N 1 | F1 | 241 |
| N N 6 5 N N 2 N | 32 | 50 | N 7 6 5 N N 2 N | 72 | 114 | 8 N 6 5 N N 2 N | B2 | 178 | 8 7 6 5 N N 2 N | F2 | 242 |
| N N 6 5 N N 2 1 | 33 | 51 | N 7 6 5 N N 2 1 | 73 | 115 | 8 N 6 5 N N 2 1 | B3 | 179 | 8 7 6 5 N N 2 1 | F3 | 243 |
| N N 6 5 N 3 N N | 34 | 52 | N 7 6 5 N 3 N N | 74 | 116 | 8 N 6 5 N 3 N N | B4 | 180 | 8 7 6 5 N 3 N N | F4 | 244 |
| N N 6 5 N 3 N 1 | 35 | 53 | N 7 6 5 N 3 N 1 | 75 | 117 | 8 N 6 5 N 3 N 1 | B5 | 181 | 8 7 6 5 N 3 N 1 | F5 | 245 |
| N N 6 5 N 3 2 N | 36 | 54 | N 7 6 5 N 3 2 N | 76 | 118 | 8 N 6 5 N 3 2 N | B6 | 182 | 8 7 6 5 N 3 2 N | F6 | 246 |
| N N 6 5 N 3 2 1 | 37 | 55 | N 7 6 5 N 3 2 1 | 77 | 119 | 8 N 6 5 N 3 2 1 | B7 | 183 | 8 7 6 5 N 3 2 1 | F7 | 247 |
| N N 6 5 4 N N N | 38 | 56 | N 7 6 5 4 N N N | 78 | 120 | 8 N 6 5 4 N N N | B8 | 184 | 8 7 6 5 4 N N N | F8 | 248 |
| N N 6 5 4 N N 1 | 39 | 57 | N 7 6 5 4 N N 1 | 79 | 121 | 8 N 6 5 4 N N 1 | B9 | 185 | 8 7 6 5 4 N N 1 | F9 | 249 |
| N N 6 5 4 N 2 N | 3A | 58 | N 7 6 5 4 N 2 N | 7A | 122 | 8 N 6 5 4 N 2 N | BA | 186 | 8 7 6 5 4 N 2 N | FA | 250 |
| N N 6 5 4 N 2 1 | 3B | 59 | N 7 6 5 4 N 2 1 | 7B | 123 | 8 N 6 5 4 N 2 1 | BB | 187 | 8 7 6 5 4 N 2 1 | FB | 251 |
| N N 6 5 4 3 N N | 3C | 60 | N 7 6 5 4 3 N N | 7C | 124 | 8 N 6 5 4 3 N N | BC | 188 | 8 7 6 5 4 3 N N | FC | 252 |
| N N 6 5 4 3 N 1 | 3D | 61 | N 7 6 5 4 3 N 1 | 7D | 125 | 8 N 6 5 4 3 N 1 | BD | 189 | 8 7 6 5 4 3 N 1 | FD | 253 |
| N N 6 5 4 3 2 N | 3E | 62 | N 7 6 5 4 3 2 N | 7E | 126 | 8 N 6 5 4 3 2 N | BE | 190 | 8 7 6 5 4 3 2 N | FE | 254 |
| N N 6 5 4 3 2 1 | 3F | 63 | N 7 6 5 4 3 2 1 | 7F | 127 | 8 N 6 5 4 3 2 1 | BF | 191 | 8 7 6 5 4 3 2 1 | FF | 255 |

Example: Decimal 146 means that inputs 8, 5 and 2 are inverted.

## 4.21 Multiplexer block

This 'Toolkit' option block selects one of eight analogue inputs to appear at its output.

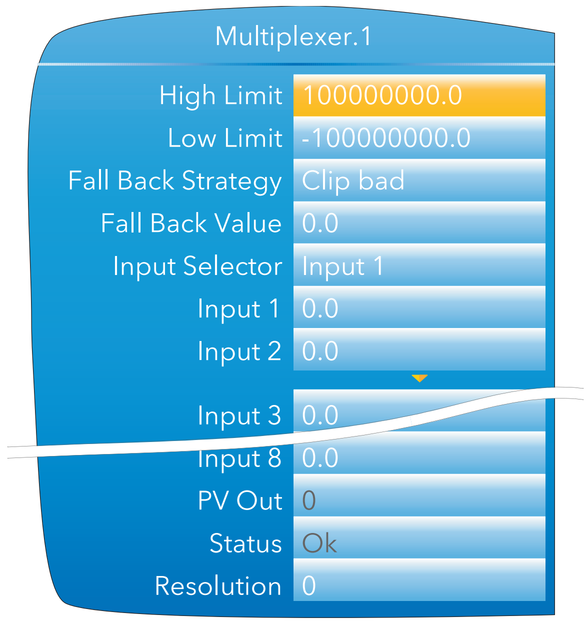



Figure 4.21 Multiplexer block configuration

### 4.21.1 Configuration parameters

| | |
|---|---|
| High Limit | The high limit for input, output and fallback values. Minimum value is Low Limit. |
| Low Limit | The low limit for input and fallback values. Maximum value is High Limit. |
| Fallback Strategy | Clip Bad: If the input value is above 'High Limit' or below 'Low Limit', then the output value is set to the appropriate limit, and the status is set to 'Bad'. If the input signal is within the limits, but its status is bad, the output is set to the Fallback value. |
| | Clip Good: If the input value is above 'High Limit' or below 'Low Limit', then the output value is set to the appropriate limit, and the status is set to 'Good'. If the input signal is within the limits, but its status is bad, the output is set to the Fallback value. |
| | Fall Bad: If the input value is above 'High Limit' or below 'Low Limit', then the output value is set to the Fallback value, and the status is set to 'Bad' |
| | Fall Good: If the input value is above 'High Limit' or below 'Low Limit', then the output value is set to the Fallback value, and the status is set to 'Good' |
| | Upscale: If the input status is bad, or if the input signal is above 'High Limit' or below 'Low Limit', the output value is set to the High limit. |
| | Downscale: If the input status is bad, or if the input signal is above 'High Limit' or below 'Low Limit', the output value is set to the Low limit. |
| Fallback Value | The value to be adopted by the output, under error conditions, if 'Fallback Status' is set to 'Fall Good' or 'Fall Bad'. |
| Input Selector | Selects which of the eight inputs is presented at the output. When wired to a suitable parameter, Input Selector becomes read only. Input 1 is selected for an Input Selector value of 1, Input 2 for a value of 2 and so on. Input Selector values greater than 8 are ignored. If not wired, the user may select the required input using the scroll keys. |
| Input 1 to 8 | Wired to the relevant analogue inputs. |
| PV Out | The output from the multiplexer block. |
| Status | Indicates the status of the operation as 'Ok' or 'Error'. |
| Resolution | The number of decimal places for the output value (maximum = 6) |

## 4.22 MATH (2 INPUT)

This 'Toolkit' option block allows one of a number of operations to be carried out using two input values which may be analogue or digital in nature. Either or both of the inputs can be scaled, using a 'Multiplier'.

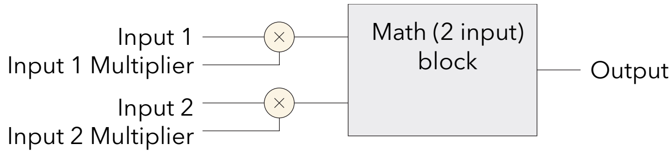



Figure 4.22a Block schematic

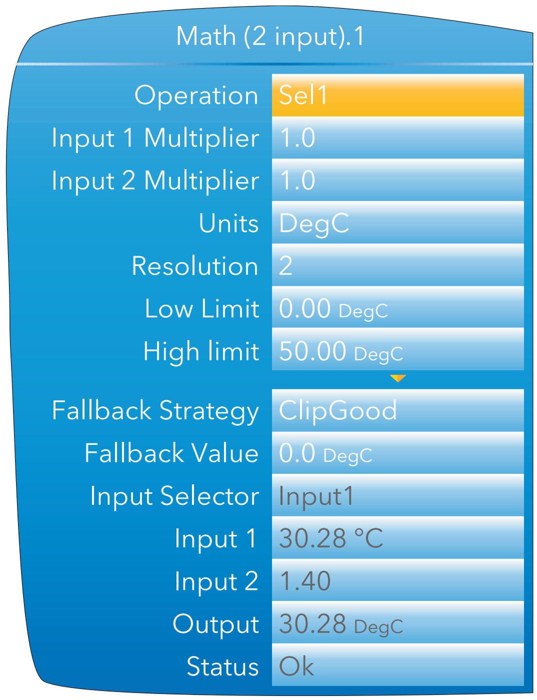



Figure 4.22b Block configuration (typical)

### 4.22.1 Parameters

| | | |
|---|---|---|
| Operation | Add | Output = Input 1 + Input 2 |
| | Subtract | Output = Input 1 - Input 2 |
| | Multiply | Output = Input 1 x Input 2 |
| | Divide | Output = Input 1 ÷ Input 2 |
| | Abs Diff | Output = the difference between Input 1 and Input 2, ignoring sign |
| | Select Max | Output = whichever is the larger of Input 1 or Input 2 |
| | Select Min | Output = whichever is the smaller of Input 1 or Input 2 |
| | Hot Swap | Output = Input 2 if Input 1 is 'Bad'; otherwise Output = Input 1 |
| | Sample/Hold | Output tracks Input 1 whilst Input 2 = 1. Output value is held whilst Input 2 = 0 (See section 4.22.2, below, for more details) |
| | Power* | Output = Input 1 to the power of Input 2. (Output = Input $1^{\text{Input 2}}$) |
| | Square Root | Output = √Input 1 (Input 2 ignored) |
| | Log Base 10 | Output = $\text{Log}_{10}$ Input 1 (Input 2 ignored) |
| | Log Base e | Output = Ln Input 1 (Input 2 ignored) |
| | Exponential | Output = $e^{\text{Input1}}$ (Input 2 ignored) |
| | 10 to the X | Output = $10^{\text{Input 1}}$ (Input 2 ignored) |
| | Sel1 | Output = Input 1 if Input Selector = Input1<br>Output = Input 2 if Input Selector = Input2 |

* Note... For this implementation:
0 to the power 0 = 1.
Negative values raised to any power result in bad status.
0 raised to a negative power results in bad status.

### 4.22.1 PARAMETERS (Cont.)

| | |
|---|---|
| Input 1(2) Multiplier | The scaling factor for input 1(2). This multiplying factor is applied to the input of the function, but does not affect the displayed values of Input1 and Input 2 (below). |
| Units | Allows a five-character string to be entered for the function |
| Resolution | Sets the number of decimal places for the Output value. Input resolution (if applicable) is that of the relevant input. |
| High Limit | The high limit for input, output and fallback values. Minimum value is Low Limit. |
| Low Limit | The low limit for input and fallback values. Maximum value is High Limit. |
| Fallback Strategy | Clip Bad: If the input value is above 'High Limit' or below 'Low Limit', then the output value is set to the appropriate limit, and the status is set to 'Bad'. If the input signal is within the limits, but its status is bad, the output is set to the Fall Back value.<br>Clip Good: If the input value is above 'High Limit' or below 'Low Limit', then the output value is set to the appropriate limit, and the status is set to 'Good'. If the input signal is within the limits, but its status is bad, the output is set to the Fall Back value.<br>Fall Bad: If the input value is above 'High Limit' or below 'Low Limit', then the output value is set to the Fall Back value, and the status is set to 'Bad'<br>Fall Good: If the input value is above 'High Limit' or below 'Low Limit', then the output value is set to the Fall Back value, and the status is set to 'Good'<br>Upscale: If the input status is bad, or if the input signal is above 'High Limit' or below 'Low Limit', the output value is set to the High limit.<br>Downscale: If the input status is bad, or if the input signal is above 'High Limit' or below 'Low Limit', the output value is set to the Low limit. |
| Fallback Value | The value to be adopted by the output, under error conditions, if 'Fallback Status' is set to 'Fall Good' or 'Fall Bad'. |
| Input Selector | For 'Select' operation only. When wired to a suitable parameter, Input Select becomes read only. Input 1 is selected if 'Input Select' = 1; Input 2 is selected if 'Input Select' = 2. Input Select values greater than 2 are ignored. If not wired, the user may select the required input using the scroll keys. |
| Input 1(2) | Wired to suitable input parameters. Displayed values ignore any input multiplier effects. |
| Output | Gives the output value for the operation. |
| Status | Shows the status of the output value, as 'Ok' or 'Error' |

### 4.22.2 Sample and Hold details

As described above, Output follows Input1 as long as Input 2 is 'High'. When Input 2 goes Low, the output adopts the instantaneous value of Input 1 until Input 2 goes High again. When Input 2 goes high the output jumps to the current value of Input 1 and tracks it until Input 2 goes low.

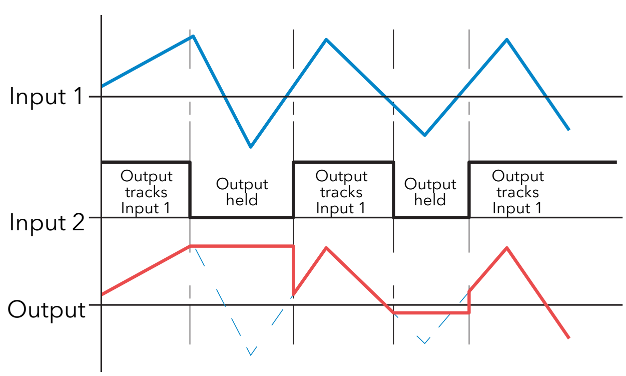



Figure 4.22.2 Sample and Hold example

## 4.23 TIMER

This 'Toolkit' option allows the user to configure up to four timers as: 'On Pulse', 'On Delay', 'One Shot' or 'Min On' types. The different types are described in section 4.23.2, below.

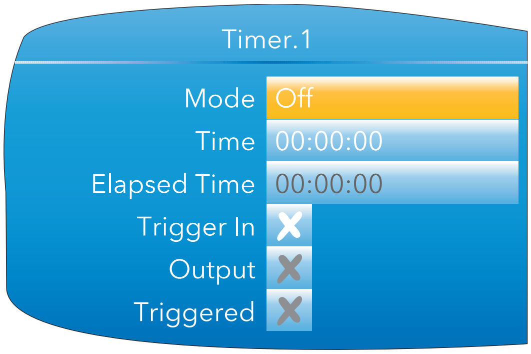



Figure 4.23 Timer configuration

### 4.23.1 Parameters

| | |
|---|---|
| Mode | Select 'On pulse', 'On delay', 'One shot' or 'Min On' |
| Time | Allows the user to enter a period for the timer. |
| Elapsed time | This read-only parameter shows timing progress |
| Trigger in | Shows if the trigger source is active (tick) or inactive (cross) |
| Output | Shows if the output is on (tick) or off (cross) |
| Triggered | Shows if the timer is currently triggered (can remain triggered even after the trigger source has returned to off). |

### 4.23.2 Timer modes

#### ON PULSE

Output goes 'on' as soon as the trigger input goes active, and remains on until the time period has elapsed. If the timer is re-triggered during the timing period, the timer restarts.

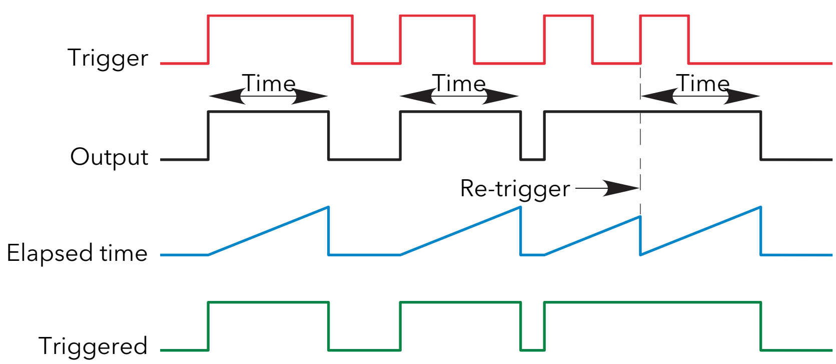



Figure 4.23.2a 'On Pulse' definitions

### 4.23.2 TIMER MODES (Cont.)

#### ON DELAY

Provides a delay between the trigger point and the timer output becoming active.

Rules

1. After the trigger goes active, the output switches on after the delay time has elapsed, and stays on until the trigger goes inactive.
2. If the trigger goes inactive before the delay time has elapsed, the output does not switch on.

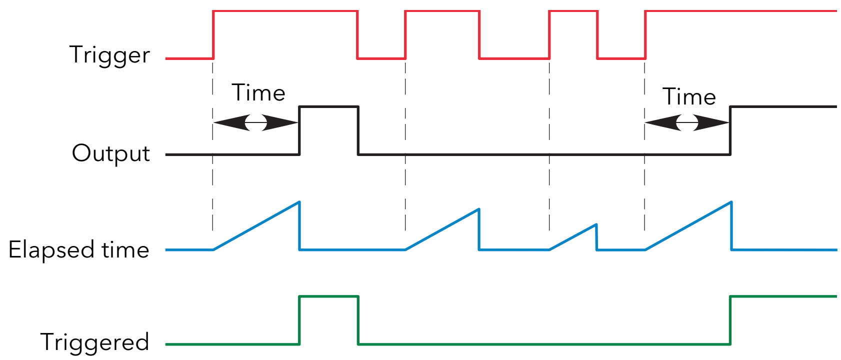



Figure 4.23.2b 'On Delay' definitions

#### ONE SHOT

If the trigger input is active, countdown timing is initiated as soon as the entered time value is confirmed (scroll key). The entered time decrements to zero, and must be re-entered by the user before any further timer function can be initiated.

Rules

1. The time value decrements only when the trigger input is active.
2. The output is On only when the trigger value is active (and the entered time value has not elapsed).
3. The entered time value can be edited at any time to increase or decrease the remaining time period.

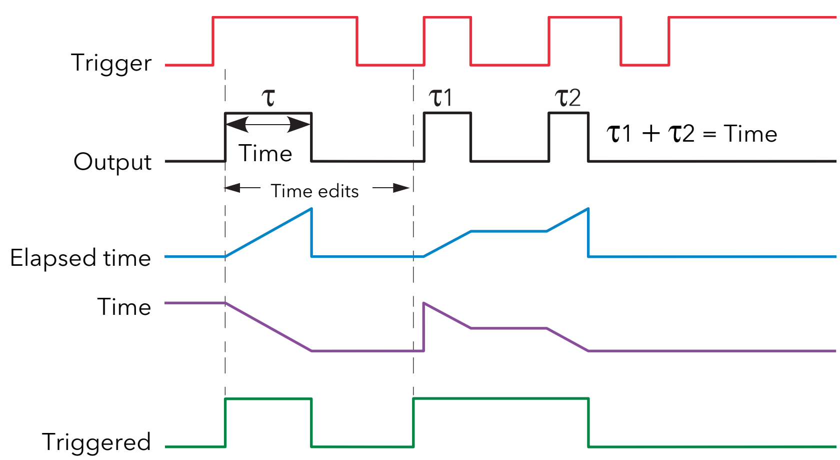



Figure 4.23.2c 'One Shot' timer definitions

Note: For ease of comparison the two time edits in the figure above were both to the same value. This is not a necessary condition.

### 4.23.2 TIMER MODES (Cont.)

#### MIN ON

This 'Off delay' function provides an output signal that goes 'on' when the trigger goes active and remains on for a specified period after the trigger goes inactive.

If the trigger goes inactive, then active again before the time period has elapsed, then the elapsed time is reset to zero and the output remains on.

The 'Triggered' parameter is on whenever the elapsed time is counting down.

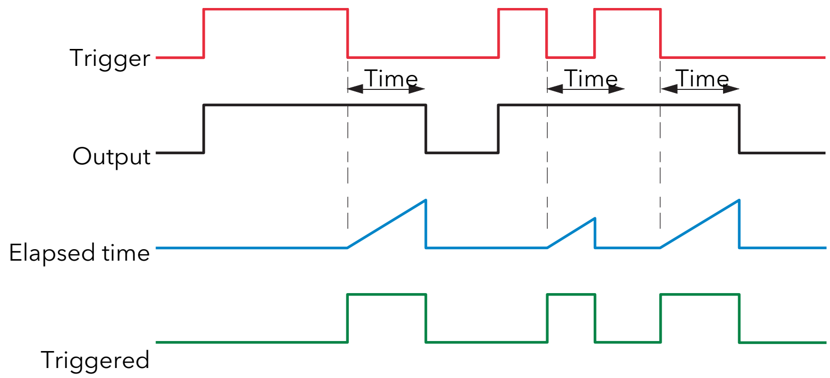



Figure 4.23.2d 'Min On' timer definitions

## 4.24 USER VALUES

This 'Toolkit' option block allows up to 12 values to be configured for use as inputs to other parameters.

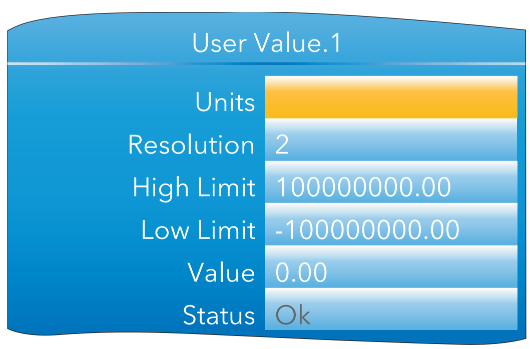



Figure 4.24 User value configuration

### 4.24.1 Parameters

| | |
|---|---|
| Units | Allows a five-character string to be entered for the user value units |
| Resolution | The number of decimal places for the user value (max. = 6) |
| High/Low Limit | Sets maximum and minimum values that the User value can be set to |
| Value | The user value, either entered manually, or wired to another appropriate parameter |
| Status | The output status for the User Value. |

## 4.25 ALARM SUMMARY

Allows the user to view the overall status of the unit's alarms, and to carry out a global acknowledgement of active alarms if required.

| | |
|---|---|
| Global Ack | Allows the user to acknowledge all applicable alarms simultaneously. 'Manual' alarms must be non-active before they can be acknowledged. |
| Any Channel alarm | Indicates if there are any channel alarms active, acknowledged etc. |
| Any Sys Alarm | Indicates if there are any active system alarms. |
| Any Alarm | Indicates if there are any channel or system alarms active. |

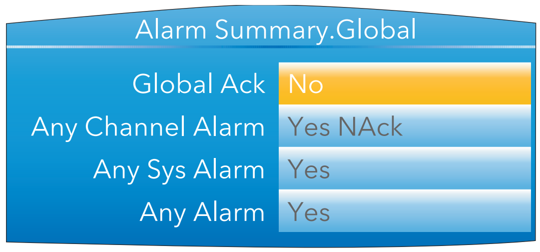



Figure 4.25 Alarm summary display

## 4.26 REAL TIME EVENT CONFIGURATION

This allows the user to configure up to two events to trigger at a specific time and date, or on a particular day, and to remain active for a configurable time, either measured as a duration, or as a specific 'Off' time.

Typical applications would be to start and/or stop a programmer at a particular time, or to act as an input to a 'Wait' segment.

Figure 4.26 shows the two types of timer: 'Time and Date', and 'Time and Day', for Event 1.


Figure 4.26 Real Time Events (typical)

| | |
|---|---|
| Type | Selects the type of the real time event (Off, Time and Day, Time and Date |
| On Month | For 'Time and Date' only, this is the month that the event is to switch on. |
| On Date | For 'Time and Date' only, this is the date in the month that the event is to switch on. |
| On Day | For 'Time and Day' only, this is the day(s) of the week that the event output is to switch on (Mon, Tue, Wed, Thu, Fri, Sat, Sun, Mon-Fri, Sat-Sun, Everyday). |
| On Time | The time of day that the event output is to switch on (00:00:00 to 23:59:59) |
| Off Type | Selects the action that will switch the event off (Duration, Time) |
| Off Month | For 'Time and Date' only and with 'Off Type' set to 'Time', this is the month that the event is to switch off. |
| Off Date | For 'Time and Date' only and with 'Off Type' set to 'Time', this is the day number in the month that the event is to switch off. |
| Off Day | For 'Time and Day' only and with 'Off Type' set to 'Time', this is the day of the week that the event output is to switch off (Mon, Tue, Wed, Thu, Fri, Sat, Sun, Mon-Fri, Sat-Sun, Everyday). |
| Off Time | The time at which the event output is to switch off (00:00:00 - 23:59:59) |
| Duration | For 'Off type' set to 'Duration', this specifies the duration for which the event output is to remain on (00:00:01 to 23:59:59 for Time and Day, or 00:00:01 to 500:00:00 for Time and Date) |
| Output | The output for the real time event (Cross symbol = Off, Tick = On) (Read only) |

# 5 MODBUS TCP SLAVE COMMS

## 5.1 INSTALLATION

The installation of the Modbus link consists of connecting a standard Ethernet cable between the RJ45 connector at the rear of the unit to a host computer either directly or via a network. A 'straight-through' cable can be used in either case (i.e. a cross-over cable is not required).

## 5.2 INTRODUCTION

MODBUS TCP allows the instrument to act as a 'slave' device to one or more host computers connected via the RJ45 connector at the rear of the recorder. Each recorder must have a unique Internet Protocol (IP) address, set up as described in Section 4.2.1 (Network.Interface).

MODBUS TCP (Transmission Control Protocol) is a variant of the MODBUS family of communications protocols intended for supervision and control of automated equipment specifically covering the use of MODBUS messaging in an intranet or internet environment, using TCP/IP protocols. Much of the MODBUS detail in this manual is derived from the document openmbus.doc, available at http://www.modbus.org/default.htm The above mentioned document also includes implementation guidelines for users.

Note: The Modbus protocol allows a maximum of 255 data bytes to be read from or written to in one transaction. For this reason, the maximum number of standard (16 bit) registers that can be accessed in one transaction is 255/2 = 127 and the maximum number of IEEE (32-bit) registers is 127/2 = 63.

### 5.2.1 Function Codes

MODBUS function codes 3, 4, 6, 8 and 16, defined in table 8.2.1a below, are supported and are fully described in section 5.5, below.

| Code | Modbus definition | Description |
|---|---|---|
| 03 | Read holding registers | Reads the binary contents if holding registers. In this implementation codes 3 and 4 are identical in operation. |
| 04 | Read input registers | Reads the binary contents if holding registers. In this implementation codes 3 and 4 are identical in operation. |
| 06 | Preset single register | Writes a single value to a single register. |
| 08 | Diagnostics | Performs a simple loop back test. |
| 16 | Preset multiple registers | Writes values to multiple holding registers. |

Table 5.2.1a MODBUS Function code definition

**DIAGNOSTIC CODES**

Function code 08, subfunction 00 (Return query data) echoes the query (Loop back).

### 5.2.1 FUNCTION CODES (Cont.)

#### EXCEPTION CODES

MODBUS TCP provides reserved codes used for exceptions. These codes provide error information relating to failed requests. Exceptions are signalled by hex 80 being added to the function code of the request, followed by one of the codes listed in table 8.2.1b, below.

| Code Dec | Code Hex | Modbus definition | Description (see Modbus specification for full details) |
|---|---|---|---|
| 01 | 01 | Illegal function | An invalid function code was received |
| 02 | 02 | Illegal Data Address | An invalid data address was received |
| 03 | 03 | Illegal Data Value | An invalid data value was received |
| 04 | 04 | Slave Device Failure | An unrecoverable error occurred in the instrument |
| 09 | 09 | Illegal Sub Function | An invalid sub function was received |
| 10 | 0A | Gateway path unavailable | Gateway misconfigured or overloaded |
| 11 | 0B | Gateway target device failed to respond | Device not present on the network |

Table 5.2.1b Exception codes

## 5.2.2 Data types

The following data types are supported:

1. 2's complement signed 16-bit analogue values with implied decimal point. The decimal point position must be configured in both the recorder and the host computer.
2. 16, 32 and 64 bit signed integers.
3. 16-bit unsigned integer values.
4. 32 bit IEEE Floating point values.
5. Strings of limited size, can be transferred across Modbus TCP in Unicode format using a single non-multiplexed set of consecutive registers.

#### DATA ENCODING

MODBUS uses what is called a 'Big endian' representation for addresses and data items. This means that when a numerical quantity larger than a single byte is transmitted, the most significant byte is sent first. For example a 32-bit hex value of 12345678 would be transmitted as 12, followed by 34, followed by 56 and finally 78.

## 5.2.3 Invalid multiple register writes

When a recorder receives a multi-register write request, it is possible that one or more requests will be rejected. Under such a circumstance, the recorder accepts all valid write requests and ignores any invalid writes. No error response is produced.

## 5.2.4 Master communications timeout

Whilst the instrument is archiving, it is possible that communications responses slow sufficiently to cause communications timouts. The Modbus master device should be configured with a timout value large enough to ensure against nuisance timeouts during archiving.

### 5.2.4 Non-volatile parameters in EEPROM

**CAUTION**

The parameters in the following list must not be written-to on a continuous basis as to do so will damage the EEPROM, greatly shortening its useful life.

Note: 'nvol' = 'non-volatile'. Loop 'N' = Loop1 and Loop2; Channel 'N' = Channel 1, 2, 3 and 4 etc.

AdvancedLoop.MasterPID.ControlAction
AdvancedLoop.MasterPID.CutbackHigh
AdvancedLoop.MasterPID.CutbackLow
AdvancedLoop.MasterPID.DerivativeTime
AdvancedLoop.MasterPID.DerivativeType
AdvancedLoop.MasterPID.ErrorLimit
AdvancedLoop.MasterPID.IntegralTime
AdvancedLoop.MasterPID.LoopBreakTime
AdvancedLoop.MasterPID.ManualReset
AdvancedLoop.MasterPID.PBUnits
AdvancedLoop.MasterPID.ProportionalBand
AdvancedLoop.MasterSP.ManualTrack
AdvancedLoop.MasterSP.RangeHigh
AdvancedLoop.MasterSP.RangeLow
AdvancedLoop.MasterSP.ServoToPV
AdvancedLoop.MasterSP.SPHighLimit
AdvancedLoop.MasterSP.SPIntBal
AdvancedLoop.MasterSP.SPLowLimit
AdvancedLoop.MasterSP.SPTrack
AdvancedLoop.MasterSP.SPTrimHighLimit
AdvancedLoop.MasterSP.SPTrimLowLimit
AdvancedLoop.Output.Ch1OnOffHysteresis
AdvancedLoop.Output.Ch1TravelTime
AdvancedLoop.Output.Ch2Deadband
AdvancedLoop.Output.Ch2OnOffHysteresis
AdvancedLoop.Output.Ch2TravelTime
AdvancedLoop.Output.CoolType
AdvancedLoop.Output.EnablePowerFeedforward
AdvancedLoop.Output.FeedForwardGain
AdvancedLoop.Output.FeedForwardOffset
AdvancedLoop.Output.FeedForwardTrimLimit
AdvancedLoop.Output.FeedForwardType
AdvancedLoop.Output.ManualMode
AdvancedLoop.Output.ManualStartup
AdvancedLoop.Output.OutputHighLimit
AdvancedLoop.Output.OutputLowLimit
AdvancedLoop.Output.PotBreakMode
AdvancedLoop.Output.Rate
AdvancedLoop.Output.RateDisable
AdvancedLoop.Output.SafeOutVal
AdvancedLoop.Output.SbrkOP
AdvancedLoop.Output.SlaveSensorBreakMode
AdvancedLoop.Setup.CascadeType
AdvancedLoop.Setup.MasterName
AdvancedLoop.Setup.ModeAccess
AdvancedLoop.Setup.SetpointAccess
AdvancedLoop.Setup.SlaveChannel1
AdvancedLoop.Setup.SlaveChannel2
AdvancedLoop.Setup.SlaveName
AdvancedLoop.SlavePID.Boundary1-2
AdvancedLoop.SlavePID.Boundary2-3
AdvancedLoop.SlavePID.ControlAction
AdvancedLoop.SlavePID.CutbackHigh
AdvancedLoop.SlavePID.CutbackHigh2
AdvancedLoop.SlavePID.CutbackHigh3
AdvancedLoop.SlavePID.CutbackLow
AdvancedLoop.SlavePID.CutbackLow2
AdvancedLoop.SlavePID.CutbackLow3
AdvancedLoop.SlavePID.DerivativeTime
AdvancedLoop.SlavePID.DerivativeTime2
AdvancedLoop.SlavePID.DerivativeTime3
AdvancedLoop.SlavePID.DerivativeType
AdvancedLoop.SlavePID.IntegralTime
AdvancedLoop.SlavePID.IntegralTime2
AdvancedLoop.SlavePID.IntegralTime3
AdvancedLoop.SlavePID.LoopBreakTime
AdvancedLoop.SlavePID.LoopBreakTime2
AdvancedLoop.SlavePID.LoopBreakTime3
AdvancedLoop.SlavePID.ManualReset
AdvancedLoop.SlavePID.ManualReset2
AdvancedLoop.SlavePID.ManualReset3
AdvancedLoop.SlavePID.NumberOfSets
AdvancedLoop.SlavePID.PBUnits
AdvancedLoop.SlavePID.ProportionalBand
AdvancedLoop.SlavePID.ProportionalBand2
AdvancedLoop.SlavePID.ProportionalBand3
AdvancedLoop.SlavePID.RelCh2Gain
AdvancedLoop.SlavePID.RelCh2Gain2
AdvancedLoop.SlavePID.RelCh2Gain3
AdvancedLoop.SlavePID.RemoteInput
AdvancedLoop.SlavePID.SchedulerType
AdvancedLoop.SlaveSP.FFSelect
AdvancedLoop.SlaveSP.ManualTrack
AdvancedLoop.SlaveSP.MasterSensorBreakMode
AdvancedLoop.SlaveSP.RangeHigh
AdvancedLoop.SlaveSP.RangeLow
AdvancedLoop.SlaveSP.SbrkSP
AdvancedLoop.Tune.Band
AdvancedLoop.Tune.CycleNo
AdvancedLoop.Tune.Hysteresis
AdvancedLoop.Tune.OutputHighLimit
AdvancedLoop.Tune.OutputLowLimit
AdvancedLoop.Tune.PBs
AdvancedLoop.Tune.Settle
AdvancedLoop.Tune.TDs
AdvancedLoop.Tune.Timeout
AdvancedLoop.Tune.TuneHigh
AdvancedLoop.Tune.TuneLow
AdvancedLoop.Tune.TuneR2G
AdvancedLoop.Tune.TuneType
BCDInput.N.InN
Channel.N.AlarmN.Amount
Channel.N.AlarmN.AverageTime
Channel.N.AlarmN.Block
Channel.N.AlarmN.ChangeTime
Channel.N.AlarmN.Deviation
Channel.N.AlarmN.Dwell
Channel.N.AlarmN.Hysteresis
Channel.N.AlarmN.Latch
Channel.N.AlarmN.Threshold
Channel.N.AlarmN.Type
Channel.N.Main.CJType
Channel.N.Main.CloseString
Channel.N.Main.Descriptor
Channel.N.Main.ExtCJTemp
Channel.N.Main.FaultResponse
Channel.N.Main.Filter
Channel.N.Main.InputHigh
Channel.N.Main.InputLow
Channel.N.Main.LinType
Channel.N.Main.Offset
Channel.N.Main.Offset2
Channel.N.Main.OpenString
Channel.N.Main.RangeHigh
Channel.N.Main.RangeLow
Channel.N.Main.RangeUnits
Channel.N.Main.Resolution
Channel.N.Main.ScaleHigh
Channel.N.Main.ScaleHigh2
Channel.N.Main.ScaleLow
Channel.N.Main.ScaleLow2
Channel.N.Main.SensorBreakType
Channel.N.Main.Shunt
Channel.N.Main.TestSignal
Channel.N.Main.Type
Channel.N.Main.Units
Channel.N.Trend.Colour
Channel.N.Trend.SpanHigh
Channel.N.Trend.SpanLow
CustomMessage.MessageN
DCOutput.1A1B_DCOP.FallbackPV
DCOutput.1A1B_DCOP.OutputHigh
DCOutput.1A1B_DCOP.OutputLow
DCOutput.1A1B_DCOP.Resolution
DCOutput.1A1B_DCOP.ScaleHigh
DCOutput.1A1B_DCOP.ScaleLow
DCOutput.1A1B_DCOP.Type
DCOutput.2A2B_DCOP.FallbackPV
DCOutput.2A2B_DCOP.OutputHigh
DCOutput.2A2B_DCOP.OutputLow
DCOutput.2A2B_DCOP.Resolution
DCOutput.2A2B_DCOP.ScaleHigh
DCOutput.2A2B_DCOP.ScaleLow
DCOutput.2A2B_DCOP.Type
DCOutput.3A3B_DCOP.FallbackPV
DCOutput.3A3B_DCOP.OutputHigh
DCOutput.3A3B_DCOP.OutputLow
DCOutput.3A3B_DCOP.Resolution
DCOutput.3A3B_DCOP.ScaleHigh
DCOutput.3A3B_DCOP.ScaleLow
DCOutput.3A3B_DCOP.Type
DigitalIO.1A1B.Backlash
DigitalIO.1A1B.Inertia
DigitalIO.1A1B.Invert
DigitalIO.1A1B.MinOnTime
DigitalIO.1A1B.StandbyAction
DigitalIO.1A1B.Type
DigitalIO.2A2B.Backlash
DigitalIO.2A2B.Inertia
DigitalIO.2A2B.Invert
DigitalIO.2A2B.MinOnTime
DigitalIO.2A2B.StandbyAction
DigitalIO.2A2B.Type
DigitalIO.3A3B.Backlash
DigitalIO.3A3B.Inertia
DigitalIO.3A3B.Invert
DigitalIO.3A3B.MinOnTime
DigitalIO.3A3B.StandbyAction
DigitalIO.3A3B.Type
DigitalIO.DI_LALC.Backlash
DigitalIO.DI_LALC.Inertia
DigitalIO.DI_LALC.Invert
DigitalIO.DI_LALC.MinOnTime
DigitalIO.DI_LALC.StandbyAction
DigitalIO.DI_LALC.Type
DigitalIO.DI_LBLC.Backlash
DigitalIO.DI_LBLC.Inertia
DigitalIO.DI_LBLC.Invert
DigitalIO.DI_LBLC.MinOnTime
DigitalIO.DI_LBLC.StandbyAction
DigitalIO.DI_LBLC.Type
DigitalIO.RELAY_4AC.Backlash
DigitalIO.RELAY_4AC.Inertia
DigitalIO.RELAY_4AC.Invert
DigitalIO.RELAY_4AC.MinOnTime
DigitalIO.RELAY_4AC.StandbyAction
DigitalIO.RELAY_4AC.Type
DigitalIO.RELAY_5AC.Backlash
DigitalIO.RELAY_5AC.Inertia
DigitalIO.RELAY_5AC.Invert
DigitalIO.RELAY_5AC.MinOnTime
DigitalIO.RELAY_5AC.StandbyAction
DigitalIO.RELAY_5AC.Type
EthernetIP.ImplicitInputs.InputN
EthernetIP.ImplicitOutputs.OutputN
EthernetIP.InputTags.InputN
EthernetIP.Main.ConfigInstance
EthernetIP.Main.ConfigSize
EthernetIP.Main.ConnectionType
EthernetIP.Main.InputInstance
EthernetIP.Main.InputSize
EthernetIP.Main.Mode
EthernetIP.Main.OutputInstance
EthernetIP.Main.OutputSize
EthernetIP.Main.Priority
EthernetIP.Main.Rpi
EthernetIP.Main.ServerAddress
EthernetIP.Main.SlotNumber
EthernetIP.OutputTags.OutputN
EthernetIP.OutputTags.Output2
Group.Recording.ChannelNEn
Group.Recording.Compression
Group.Recording.Enable
Group.Recording.Interval
Group.Recording.VirtualChanNEn
Group.Recording.VirtualChan28En
Group.Trend.Descriptor
Group.Trend.Interval
Group.Trend.MajorDivisions
Group.Trend.PointN
Humidity.Pressure
Humidity.PsychroConst
Humidity.Resolution
Humidity.WetOffset
Instrument.Display.AlarmPanel
Instrument.Display.Brightness
Instrument.Display.Cascade
Instrument.Display.DualLoopControl
Instrument.Display.EIPServerPage
Instrument.Display.FutureTrend
Instrument.Display.FutureTrend1Colour
Instrument.Display.FutureTrend2Colour
Instrument.Display.HistoryBackground
Instrument.Display.HomePage
Instrument.Display.HorizontalBar
Instrument.Display.HorizontalTrend
Instrument.Display.HPageTimeout
Instrument.Display.HTrendScaling
Instrument.Display.LoopControl
Instrument.Display.LoopSetpointColour
Instrument.Display.ModbusMaster
Instrument.Display.NumberFormat

### 5.2.4 NON-VOLATILE PARAMETERS IN EEPROM (Cont.)

Instrument.Display.Numeric
Instrument.Display.Programmer
Instrument.Display.PromoteListView
Instrument.Display.ScreenSaverAfter
Instrument.Display.ScreenSaverBrightness
Instrument.Display.SteriliserPage
Instrument.Display.TrendBackground
Instrument.Display.USBAutoScan
Instrument.Display.VerticalBar
Instrument.Display.VerticalTrend
Instrument.Info.CloneState
Instrument.Info.Name
Instrument.Locale.DateFormat
Instrument.Locale.DSTenable
Instrument.Locale.EndDay
Instrument.Locale.EndMonth
Instrument.Locale.EndOn
Instrument.Locale.EndTime
Instrument.Locale.Language
Instrument.Locale.StartDay
Instrument.Locale.StartMonth
Instrument.Locale.StartOn
Instrument.Locale.StartTime
Instrument.Locale.TimeZone
Instrument.Notes.NoteN
Instrument.PromoteList.PromoteListName
Instrument.PromoteList.PromoteParamN
Instrument.PromoteList.PromoteParamNDesc
Instrument.Security.CommsPass
Instrument.Security.DefaultConfig
Instrument.Security.EngineerPassword
Instrument.Security.OEMPass
Instrument.Security.OperatorPassword
Instrument.Security.SupervisorPassword
Lgc2.N.FallbackType
Lgc2.N.In1
Lgc2.N.In2
Lgc2.N.Invert
Lgc2.N.Oper
Lgc8.N.InN
Lgc8.N.InInvert
Lgc8.N.NumIn
Lgc8.N.Oper
Lgc8.N.OutInvert
Loop.N.Diag.LoopMode
Loop.N.OP.Ch1OnOffHysteresis
Loop.N.OP.Ch1TravelTime
Loop.N.OP.Ch2Deadband
Loop.N.OP.Ch2OnOffHysteresis
Loop.N.OP.Ch2TravelTime
Loop.N.OP.CoolType
Loop.N.OP.EnablePowerFeedforward
Loop.N.OP.FeedForwardGain
Loop.N.OP.FeedForwardOffset
Loop.N.OP.FeedForwardTrimLimit
Loop.N.OP.FeedForwardType
Loop.N.OP.ManStartup
Loop.N.OP.ManualMode
Loop.N.OP.OutputHighLimit
Loop.N.OP.OutputLowLimit
Loop.N.OP.PotBreakMode
Loop.N.OP.Rate
Loop.N.OP.RateDisable
Loop.N.OP.SafeOutVal
Loop.N.OP.SbrkOP
Loop.N.OP.SensorBreakMode
Loop.N.PID.Boundary1-2
Loop.N.PID.Boundary2-3
Loop.N.PID.CutbackHigh
Loop.N.PID.CutbackHighN
Loop.N.PID.CutbackLow
Loop.N.PID.CutbackLowN
Loop.N.PID.DerivativeTime
Loop.N.PID.DerivativeTimeN
Loop.N.PID.IntegralTime
Loop.N.PID.IntegralTimeN
Loop.N.PID.LoopBreakTime
Loop.N.PID.LoopBreakTimeN
Loop.N.PID.ManualReset
Loop.N.PID.ManualResetN
Loop.N.PID.NumSets
Loop.N.PID.ProportionalBand
Loop.N.PID.ProportionalBandN
Loop.N.PID.RelCh2Gain
Loop.N.PID.RelCh2GainN
Loop.N.PID.SchedulerRemoteInput
Loop.N.PID.SchedulerType
Loop.N.Setup.AutoManAccess
Loop.N.Setup.CH1ControlType
Loop.N.Setup.CH2ControlType
Loop.N.Setup.ControlAction
Loop.N.Setup.DerivativeType
Loop.N.Setup.LoopName
Loop.N.Setup.PBUnits
Loop.N.Setup.SPAccess
Loop.N.SP.ManualTrack
Loop.N.SP.RangeHigh
Loop.N.SP.RangeLow
Loop.N.SP.ServoToPV
Loop.N.SP.SPHighLimit
Loop.N.SP.SPIntBal
Loop.N.SP.SPLowLimit
Loop.N.SP.SPTrack
Loop.N.SP.SPTrimHighLimit
Loop.N.SP.SPTrimLowLimit
Loop.N.Tune.CycleNo
Loop.N.Tune.Diagnostics
Loop.N.Tune.OutputHighLimit
Loop.N.Tune.OutputLowLimit
Loop.N.Tune.PBs
Loop.N.Tune.Settle
Loop.N.Tune.TDs
Loop.N.Tune.TuneR2G
Loop.N.Tune.Type
Math2.N.Fallback
Math2.N.FallbackVal
Math2.N.HighLimit
Math2.N.InN
Math2.N.InNMul
Math2.N.LowLimit
Math2.N.Oper
Math2.N.Resolution
Math2.N.Select
Math2.N.Units
ModbusMaster.N.Data.BitPosition
ModbusMaster.N.Data.DataType
ModbusMaster.N.Data.Descriptor
ModbusMaster.N.Data.FallBackValue
ModbusMaster.N.Data.FunctionCode
ModbusMaster.N.Data.ModbusAddress
ModbusMaster.N.Data.Mode
ModbusMaster.N.Data.Number
ModbusMaster.N.Data.ParameterList
ModbusMaster.N.Data.Priority
ModbusMaster.N.Data.Scaling
ModbusMaster.N.Data.Set
ModbusMaster.N.Data.SlaveDevice
ModbusMaster.N.Data.Value
ModbusMaster.SlaveN.Data.BitPosition
ModbusMaster.SlaveN.Data.DataType
ModbusMaster.SlaveN.Data.Descriptor
ModbusMaster.SlaveN.Data.FallBackValue
ModbusMaster.SlaveN.Data.FunctionCode
ModbusMaster.SlaveN.Data.ModbusAddress
ModbusMaster.SlaveN.Data.Mode
ModbusMaster.SlaveN.Data.Number
ModbusMaster.SlaveN.Data.ParameterList
ModbusMaster.SlaveN.Data.Priority
ModbusMaster.SlaveN.Data.Scaling
ModbusMaster.SlaveN.Data.Set
ModbusMaster.SlaveN.Data.SlaveDevice
ModbusMaster.SlaveN.Data.Value
ModbusMaster.SlaveN.Main.Descriptor
ModbusMaster.SlaveN.Main.HighPriority
ModbusMaster.SlaveN.Main.IPAddress
ModbusMaster.SlaveN.Main.LowPriority
ModbusMaster.SlaveN.Main.MaxBlockSize
ModbusMaster.SlaveN.Main.MediumPriority
ModbusMaster.SlaveN.Main.Online
ModbusMaster.SlaveN.Main.Profile
ModbusMaster.SlaveN.Main.Retries
ModbusMaster.SlaveN.Main.Timeout
ModbusMaster.SlaveN.Main.UnitId
Mux8.N.Fallback
Mux8.N.FallbackVal
Mux8.N.HighLimit
Mux8.N.InN
Mux8.N.LowLimit
Mux8.N.Select
Network.Archive.ArchiveRate
Network.Archive.CSVDateFormat
Network.Archive.CSVHeaders
Network.Archive.CSVHeadings
Network.Archive.CSVIncludeValues
Network.Archive.CSVMessages
Network.Archive.CSVTabDelimiter
Network.Archive.Destination
Network.Archive.FileFormat
Network.Archive.OnFull
Network.Archive.Period
Network.Archive.PrimaryPassword
Network.Archive.PrimaryUser
Network.Archive.PServerIPAddress
Network.Archive.RemotePath
Network.Archive.SecondaryPassword
Network.Archive.SecondaryUser
Network.Archive.SServerIPAddress
Network.FTPserver.Password
Network.FTPserver.Username
Network.Interface.DNSserver
Network.Interface.Gateway
Network.Interface.IPaddress
Network.Interface.IPType
Network.Interface.SubnetMask
Network.Modbus.Address
Network.Modbus.InputTimeout
Network.Modbus.PrefMasterIP
Network.Modbus.SerialMode
Network.Modbus.TimeFormat
Network.Modbus.UnitIdEnable
Program.ChNHoldback
Program.ChNHoldbackVal
Program.ChNRampUnits
Program.HoldbackStyle
Program.RampStyle
Programmer.Features.FTPStore
Programmer.Features.Holdback
Programmer.Features.Messages
Programmer.Features.PVEvent
Programmer.Features.UserValue
Programmer.FTP.IPAddress
Programmer.FTP.Password
Programmer.FTP.Username
Programmer.SetUp.ChNResolution
Programmer.SetUp.ChNServoTo
Programmer.SetUp.ChNUnits
Programmer.SetUp.Channels
Programmer.SetUp.MaxEvents
Programmer.SetUp.PowerFailAction
Programmer.SetUp.ProgEditAccess
Programmer.SetUp.ProgModeAccess
Programmer.SetUp.ProgStoreAccess
Programmer.SetUp.RateResolution
Programmer.SetUp.ResetCh1UserVal
Programmer.SetUp.ResetCh2UserVal
Programmer.SetUp.ResetEventN
RealTimeEvent.N.Duration
RealTimeEvent.N.OffDate
RealTimeEvent.N.OffDay
RealTimeEvent.N.OffMonth
RealTimeEvent.N.OffTime
RealTimeEvent.N.OffType
RealTimeEvent.N.OnDate
RealTimeEvent.N.OnDay
RealTimeEvent.N.OnMonth
RealTimeEvent.N.OnTime
RealTimeEvent.N.Type
Segment.N.ChNHoldback
Segment.N.ChNHoldbackVal
Segment.N.ChNPVEvent
Segment.N.ChNPVEventUse
Segment.N.ChNPVEventVal
Segment.N.ChNRate
Segment.N.ChNTime
Segment.N.ChNTSP
Segment.N.ChNUserVal
Segment.N.ChNWait
Segment.N.ChNWaitVal
Segment.N.Cycles
Segment.N.Duration
Segment.N.EndType
Segment.N.EventN
Segment.N.GoBackTo
Segment.N.SegmentName
Segment.N.Type
Segment.N.WaitFor
Steriliser.AutoCounter
Steriliser.FailureDwellN
Steriliser.FileByTag
Steriliser.FileTag
Steriliser.InputNPV
Steriliser.InputTypeN
Steriliser.IP1BandHigh
Steriliser.IP1BandLow
Steriliser.IP1TargetSP
Steriliser.IP2BandHigh
Steriliser.IP2BandLow
Steriliser.IP2TargetSP
Steriliser.IP3BandHigh
Steriliser.IP3BandLow
Steriliser.IP3TargetSP
Steriliser.IP4BandHigh
Steriliser.IP4BandLow
Steriliser.IP4TargetSP
Steriliser.LowLimit
Steriliser.MeasuredTemp
Steriliser.TargetTemperature
Steriliser.TargetTime
Steriliser.TargetTime121
Steriliser.TargetTime134
Steriliser.ZTemperatureInterval
Timer.N.In
Timer.N.Type
UserLin.N.NumberOfBreakpoints
UserLin.N.XN

### 5.2.4 NON-VOLATILE PARAMETERS IN EEPROM (Cont.)

| | | |
|---|---|---|
| UserLin.N.YN<br>UsrVal.N.HighLimit<br>UsrVal.N.LowLimit<br>UsrVal.N.Resolution<br>UsrVal.N.Units<br>VirtualChannel.N.AlarmN.Amount<br>VirtualChannel.N.AlarmN.AverageTime<br>VirtualChannel.N.AlarmN.Block<br>VirtualChannel.N.AlarmN.ChangeTime<br>VirtualChannel.N.AlarmN.Deviation<br>VirtualChannel.N.AlarmN.Dwell<br>VirtualChannel.N.AlarmN.Hysteresis<br>VirtualChannel.N.AlarmN.Latch<br>VirtualChannel.N.AlarmN.Threshold<br>VirtualChannel.N.AlarmN.Type<br>VirtualChannel.N.Main.Descriptor<br>VirtualChannel.N.Main.HighCutOff<br>VirtualChannel.N.Main.LowCutOff<br>VirtualChannel.N.Main.Operation | VirtualChannel.N.Main.Period<br>VirtualChannel.N.Main.PresetValue<br>VirtualChannel.N.Main.Resolution<br>VirutalChannel.N.Main.RolloverValue<br>VirtualChannel.N.Main.Type<br>VirtualChannel.N.Main.Units<br>VirtualChannel.N.Main.UnitsScaler<br>VirtualChannel.N.Trend.Colour<br>VirtualChannel.N.Trend.SpanHigh<br>VirtualChannel.N.Trend.SpanLow<br>Zirconia.Clean.CleanEnable<br>Zirconia.Clean.CleanFreq<br>Zirconia.Clean.CleanMaxTemp<br>Zirconia.Clean.CleanTime<br>Zirconia.Clean.MaxRcovTime<br>Zirconia.Clean.MinRcovTime<br>Zirconia.CleanFreq<br>Zirconia.CleanTime<br>Zirconia.GasRef | Zirconia.GasRefs.CO_Ideal<br>Zirconia.GasRefs.CO_Local<br>Zirconia.GasRefs.CO_RemoteEn<br>Zirconia.GasRefs.H2_Local<br>Zirconia.GasRefs.H2_RemoteEn<br>Zirconia.MaxRcovTime<br>Zirconia.MinCalTemp<br>Zirconia.MinRcovTime<br>Zirconia.NumResolution<br>Zirconia.OxygenExp<br>Zirconia.OxygenType<br>Zirconia.ProbeOffset<br>Zirconia.ProbeType<br>Zirconia.ProcFactor<br>Zirconia.RemGasEn<br>Zirconia.TempOffset<br>Zirconia.Tolerance |

## 5.3 PARAMETER LIST

This list is arranged in alphabetical block order and gives the memory address for each parameter in both hex and decimal.

The Modbus addresses, in the range 0x0001 - 0x3FFF, listed in the table below give access to the parameter values in a scaled integer format. It is possible to gain access to the parameter values in native format by using the following formula:

Native address = (scaled integer address x 2) + 0x8000

The blocks are ordered as follows:

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| AdvancedLoop.Diag.CalcOP | Calc OP | float32 | 031f | 799 | 1dp |
| AdvancedLoop.Diag.HiSatLim | HiSatLim | float32 | 0320 | 800 | 1dp |
| AdvancedLoop.Diag.LoSatLim | LoSatLim | float32 | 0321 | 801 | 1dp |
| AdvancedLoop.Diag.MasterDerivativeOutContrib | Master derivative output contribution | float32 | 0312 | 786 | 0dp |
| AdvancedLoop.Diag.MasterError | Master error | float32 | 030d | 781 | Same as AdvancedLoop.Main.MasterPV |
| AdvancedLoop.Diag.MasterFB | Master feedback | float32 | 031e | 798 | 1dp |
| AdvancedLoop.Diag.MasterIntegralOutContrib | Master integral output contribution | float32 | 0311 | 785 | 4dp |
| AdvancedLoop.Diag.MasterLoopBreakAlarm | Master loop break (0 = No; 1 = Yes) | bool | 0323 | 803 | Not applicable |
| AdvancedLoop.Diag.MasterPropOutContrib | Master loop proportional output contribution | float32 | 0310 | 784 | 0dp |
| AdvancedLoop.Diag.MasterSensorBreak | Master sensor break (0 = Off, 1 = On) | bool | 0313 | 787 | Not applicable |
| AdvancedLoop.Diag.OPPid | OPPID | float32 | 0322 | 802 | 1dp |
| AdvancedLoop.Diag.SchedCBH | Scheduled cutback high | float32 | 3195 | 12693 | 0dp |
| AdvancedLoop.Diag.SchedCBL | Scheduled cutback low | float32 | 3196 | 12694 | 0dp |
| AdvancedLoop.Diag.SchedLPBrk | Scheduled loop break time | float32 | 3198 | 12696 | 0dp |
| AdvancedLoop.Diag.SchedMR | Scheduled manual reset | float32 | 3197 | 12695 | 1dp |
| AdvancedLoop.Diag.SchedOutputHigh | Scheduled output high limit | float32 | 319a | 12698 | 1dp |
| AdvancedLoop.Diag.SchedOutputLow | Scheduled output low limit | float32 | 319b | 12699 | 1dp |
| AdvancedLoop.Diag.SchedPB | Scheduled proportional band | float32 | 3192 | 12690 | 1dp |
| AdvancedLoop.Diag.SchedR2G | Scheduled relative cool gain | float32 | 3199 | 12697 | 1dp |
| AdvancedLoop.Diag.SchedTd | Scheduled derivative time | float32 | 3194 | 12692 | 1dp |
| AdvancedLoop.Diag.SchedTi | Scheduled integral time | float32 | 3193 | 12691 | 1dp |
| AdvancedLoop.Diag.SlaveDerivativeOutContrib | Slave derivative output contribution | float32 | 031d | 797 | 0dp |
| AdvancedLoop.Diag.SlaveError | Slave error | float32 | 031a | 794 | Same as AdvancedLoop.Main.SlavePV |
| AdvancedLoop.Diag.SlaveIntegralOutContrib | Slave integral output contribution | float32 | 031c | 796 | 4dp |
| AdvancedLoop.Diag.SlaveLoopBreakAlarm | Slave loop break (0 = No; 1 = Yes) | bool | 030f | 783 | Not applicable |
| AdvancedLoop.Diag.SlavePropOutContrib | Slave loop proportional output contribution | float32 | 031b | 795 | 0dp |
| AdvancedLoop.Diag.SlaveSensorBreak | Slave sensor break (0 = Off; 1 = On) | bool | 0325 | 805 | Not applicable |
| AdvancedLoop.Diag.TargetOutput | Target output | float32 | 030e | 782 | Same as AdvancedLoop.Output.Out-putHighLimit |
| AdvancedLoop.Diag.WorkingOutputHigh | Slave output high limit | float32 | 0315 | 789 | 0dp |
| AdvancedLoop.Diag.WorkingOutputLow | Slave output low limit | float32 | 0314 | 788 | 0dp |
| AdvancedLoop.Main.ActiveOut | Working output | float32 | 0303 | 771 | Same as AdvancedLoop.Output.Out-putHighLimit |
| AdvancedLoop.Main.CascadeMode | Cascade mode (0 = Cascade; 1 = Slave; 2 = Manual) | uint8 | 0316 | 790 | Not applicable |
| AdvancedLoop.Main.Inhibit | Control inhibit (0 = No; 1 = Yes) | bool | 0304 | 772 | Not applicable |
| AdvancedLoop.Main.MasterIntHold | Master integral hold (0 = No; 1 = Yes) | uint8 | 0305 | 773 | Not applicable |
| AdvancedLoop.Main.MasterPV | Master loop process variable | float32 | 0317 | 791 | 1dp |
| AdvancedLoop.Main.MasterWSP | Master loop working setpoint | float32 | 0318 | 792 | Same as AdvancedLoop.Main.MasterPV |
| AdvancedLoop.Main.SlaveIntHold | Slave integral hold (0 = No; 1 = Yes) | uint8 | 0306 | 774 | Not applicable |
| AdvancedLoop.Main.SlavePV | Slave loop process variable | float32 | 0300 | 768 | 1dp |
| AdvancedLoop.Main.SlaveWSP | Slave loop working setpoint | float32 | 0302 | 770 | Same as AdvancedLoop.Main.SlavePV |
| AdvancedLoop.Main.TargetSetpoint | Target setpoint | float32 | 0301 | 769 | Same as AdvancedLoop.Main.MasterPV |
| AdvancedLoop.MasterPID.ControlAction | Control action (0 = Reverse acting; 1 = Direct acting) | uint8 | 3103 | 12547 | Not applicable |
| AdvancedLoop.MasterPID.CutbackHigh | Cutback high (0 = Auto) | float32 | 31af | 12719 | 1dp |
| AdvancedLoop.MasterPID.CutbackLow | Cutback low (0 = Auto) | float32 | 31b0 | 12720 | 1dp |
| AdvancedLoop.MasterPID.DerivativeTime | Derivative time (0 = Off) | float32 | 31ae | 12718 | 1dp |
| AdvancedLoop.MasterPID.DerivativeType | Derivative type ( 0 = PV; 1 = Eror) | uint8 | 3105 | 12549 | Not applicable |
| AdvancedLoop.MasterPID.ErrorLimit | Error limit | float32 | 31cc | 12748 | 1dp |
| AdvancedLoop.MasterPID.IntegralTime | Integral time (0 = Off) | float32 | 31ad | 12717 | 1dp |
| AdvancedLoop.MasterPID.LoopBreakTime | Loop break time (0 = Off) | float32 | 31b2 | 12722 | 0dp |
| AdvancedLoop.MasterPID.ManualReset | Manual reset | float32 | 31b1 | 12721 | 1dp |
| AdvancedLoop.MasterPID.PBUnits | Proportional band units (0 = Engineering; 1 = Percentage) | uint8 | 3104 | 12548 | Not applicable |
| AdvancedLoop.MasterPID.ProportionalBand | Proportional band | float32 | 31ac | 12716 | 1dp |
| AdvancedLoop.MasterSP.AltSP | Alternative setpoint | float32 | 3160 | 12640 | Same as AdvancedLoop.Main.MasterPV |
| AdvancedLoop.MasterSP.AltSPSelect | Alternative setpoint enable (0 = No; 1 = Yes) | uint8 | 3161 | 12641 | Not applicable |
| AdvancedLoop.MasterSP.ManualTrack | Manual track enable (0 = Off; 1 = On) | uint8 | 3167 | 12647 | Not applicable |
| AdvancedLoop.MasterSP.RangeHigh | Range high | float32 | 3159 | 12633 | Same as AdvancedLoop.Main.MasterPV |
| AdvancedLoop.MasterSP.RangeLow | Range low | float32 | 315a | 12634 | Same as AdvancedLoop.Main.MasterPV |
| AdvancedLoop.MasterSP.Rate | Setpoint rate limit value (0 = Off) | float32 | 3162 | 12642 | Same as AdvancedLoop.Main.MasterPV |
| AdvancedLoop.MasterSP.RateDisable | Setpoint rate limit disable (0 = No; 1 = Yes) | bool | 3163 | 12643 | Not applicable |
| AdvancedLoop.MasterSP.RateDone | Setpoint rate limit complete (0 = No; 1 = Yes) | bool | 030a | 778 | Not applicable |
| AdvancedLoop.MasterSP.ServoToPV | Servo to PV enable (0 = No; 1 = Yes) | bool | 316c | 12652 | Not applicable |
| AdvancedLoop.MasterSP.SP1 | Setpoint 1 | float32 | 315c | 12636 | Same as AdvancedLoop.Main.MasterPV |
| AdvancedLoop.MasterSP.SP2 | Setpoint 2 | float32 | 315d | 12637 | Same as AdvancedLoop.Main.MasterPV |
| AdvancedLoop.MasterSP.SPHighLimit | Setpoint high limit | float32 | 315e | 12638 | Same as AdvancedLoop.Main.MasterPV |
| AdvancedLoop.MasterSP.SPIntBal | SP integral balance (0 = Off; 1 = On) | bool | 316b | 12651 | Not applicable |
| AdvancedLoop.MasterSP.SPLowLimit | Setpoint low limit | float32 | 315f | 12639 | Same as AdvancedLoop.Main.MasterPV |
| AdvancedLoop.MasterSP.SPSelect | Active setpoint select (0 - Setpoint 1; 1 = Setpoint 2) | uint8 | 315b | 12635 | Not applicable |
| AdvancedLoop.MasterSP.SPTrack | Setpoint tracking enable (0 = Off; 1 = On) | uint8 | 3168 | 12648 | Not applicable |
| AdvancedLoop.MasterSP.SPTrim | Setpoint trim | float32 | 3164 | 12644 | Same as AdvancedLoop.Main.MasterPV |
| AdvancedLoop.MasterSP.SPTrimHighLimit | Setpoint trim high limit | float32 | 3165 | 12645 | Same as AdvancedLoop.Main.MasterPV |
| AdvancedLoop.MasterSP.SPTrimLowLimit | Setpoint trim low limit | float32 | 3166 | 12646 | Same as AdvancedLoop.Main.MasterPV |
| AdvancedLoop.MasterSP.TrackPV | Track PV | float32 | 3169 | 12649 | Same as AdvancedLoop.Main.MasterPV |
| AdvancedLoop.MasterSP.TrackSP | Track SP | float32 | 316a | 12650 | Same as AdvancedLoop.Main.MasterPV |
| AdvancedLoop.Output.Ch1OnOffHysteresis | Channel 1 on/off hysteresis | float32 | 3172 | 12658 | Same as AdvancedLoop.Main.MasterPV |
| AdvancedLoop.Output.Ch1Output | Channel 1 output value | float32 | 030b | 779 | Same as AdvancedLoop.Output.Out-putHighLimit |
| AdvancedLoop.Output.Ch1PotBreak | Channel 1 potentiometer break (0 = Off; 1 = On) | uint8 | 3179 | 12665 | Not applicable |
| AdvancedLoop.Output.Ch1PotPosition | Channel 1 valve position | float32 | 3178 | 12664 | 0dp |
| AdvancedLoop.Output.Ch1TravelTime | Channel 1 travel time | float32 | 3174 | 12660 | 1dp |
| AdvancedLoop.Output.Ch2Deadband | Channel 2 deadband (0 = Off) | float32 | 316f | 12655 | Same as AdvancedLoop.Output.Out-putHighLimit |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| AdvancedLoop.Output.Ch2OnOffHysteresis | Channel 2 on/off hysteresis | float32 | 3173 | 12659 | Same as AdvancedLoop.Main.MasterPV |
| AdvancedLoop.Output.Ch2Output | Channel 2 (cool) output value | float32 | 030c | 780 | Same as AdvancedLoop.Output.Out-putHighLimit |
| AdvancedLoop.Output.Ch2PotBreak | Channel 2 potentiometer break (0 = Off; 1 = On) | uint8 | 317b | 12667 | Not applicable |
| AdvancedLoop.Output.Ch2PotPosition | Channel 2 valve position | float32 | 317a | 12666 | 0dp |
| AdvancedLoop.Output.Ch2TravelTime | Channel 2 travel time | float32 | 3175 | 12661 | 1dp |
| AdvancedLoop.Output.CoolType | Cooling algorithm type<br>0 = Linear 1 = Oil 2 = Water 3 = Fan | uint8 | 3183 | 12675 | Not applicable |
| AdvancedLoop.Output.EnablePowerFeedforward | Power feed forward enable (0 = No; 1 = Yes) | uint8 | 3181 | 12673 | Not applicable |
| AdvancedLoop.Output.FeedForwardGain | Feedforward gain | float32 | 3185 | 12677 | 3dp |
| AdvancedLoop.Output.FeedForwardOffset | Feedforward offset | float32 | 3186 | 12678 | 0dp |
| AdvancedLoop.Output.FeedForwardOutput | Feedforward output | float32 | 3188 | 12680 | 0dp |
| AdvancedLoop.Output.FeedForwardRemote | Feedforward remote | float32 | 318d | 12685 | 0dp |
| AdvancedLoop.Output.FeedForwardTrimLimit | Feedforward trim limit | float32 | 3187 | 12679 | 0dp |
| AdvancedLoop.Output.FeedForwardType | Feedforward type<br>0 = None 1 = Remote 2 = SP 3 = PV | uint8 | 3184 | 12676 | Not applicable |
| AdvancedLoop.Output.ForcedOP | Forced manual output value | float32 | 318f | 12687 | 1dp |
| AdvancedLoop.Output.ManualMode | Manual output mode (0 = Track; 1 = Step; 2 = LastMOP) | uint8 | 317f | 12671 | Not applicable |
| AdvancedLoop.Output.ManualOutVal | Manual output value | float32 | 3180 | 12672 | Same as AdvancedLoop.Output.Out-putHighLimit |
| AdvancedLoop.Output.ManualStartup | Manual startup mode (0 = Off; 1 = On) | bool | 3190 | 12688 | Not applicable |
| AdvancedLoop.Output.MeasuredPower | Measured mains voltage | float32 | 3182 | 12674 | 0dp |
| AdvancedLoop.Output.NudgeLower | Valve nudge lower (0 = No; 1 = Yes) | uint8 | 3177 | 12663 | Not applicable |
| AdvancedLoop.Output.NudgeRaise | Valve nudge raise (0 = No; 1 = Yes) | uint8 | 3176 | 12662 | Not applicable |
| AdvancedLoop.Output.OutputHighLimit | Output high limit | float32 | 316d | 12653 | 1dp |
| AdvancedLoop.Output.OutputLowLimit | Output low limit | float32 | 316e | 12654 | Same as AdvancedLoop.Output.Out-putHighLimit |
| AdvancedLoop.Output.PotBreakMode | Potentiometer break mode<br>0 = Raise 1 = Lower 2 = Reset 3 = Model | uint8 | 317c | 12668 | Not applicable |
| AdvancedLoop.Output.Rate | Output rate limit value (0 = Off) | float32 | 3170 | 12656 | 1dp |
| AdvancedLoop.Output.RateDisable | Rate disable (0 = No; 1 = Yes) | bool | 3171 | 12657 | Not applicable |
| AdvancedLoop.Output.RemoteOutputHigh | Remote output high limit | float32 | 318c | 12684 | Same as AdvancedLoop.Main.ActiveOut |
| AdvancedLoop.Output.RemoteOutputLow | Remote output low limit | float32 | 318b | 12683 | Same as AdvancedLoop.Main.ActiveOut |
| AdvancedLoop.Output.SafeOutVal | Safe output value | float32 | 317e | 12670 | Same as AdvancedLoop.Output.Out-putHighLimit |
| AdvancedLoop.Output.SbrkOP | Sensor break output | float32 | 318e | 12686 | Same as AdvancedLoop.Output.Out-putHighLimit |
| AdvancedLoop.Output.SlaveSensorBreakMode | Slave sensor break mode (0 = SbrkOP; 1 = Hold) | uint8 | 317d | 12669 | Not applicable |
| AdvancedLoop.Output.TrackEnable | Enable output tracking (0 = Off; 1 = On) | uint8 | 318a | 12682 | Not applicable |
| AdvancedLoop.Output.TrackOutput | Output track value | float32 | 3189 | 12681 | 0dp |
| AdvancedLoop.Setup.CascadeType | Cascade type (0 = Full scale; 1 = Trim) | uint8 | 1606 | 5638 | Not applicable |
| AdvancedLoop.Setup.MasterLoop | Master loop type (0 = PID) | uint8 | 31b3 | 12723 | Not applicable |
| AdvancedLoop.Setup.MasterName | Master loop name | string_t | 7010 | 28688 | Not applicable |
| AdvancedLoop.Setup.ModeAccess | Mode access<br>0 = R/W (Logged out) 1 = R/W (Operator) 2 = Read Only | uint8 | 31a8 | 12712 | Not applicable |
| AdvancedLoop.Setup.SetpointAccess | Setpoint access (as Mode Access, above) | uint8 | 31a7 | 12711 | Not applicable |
| AdvancedLoop.Setup.SlaveChannel1 | Slave heat/channel 1 control type<br>0 = Off 1 = On/Off 2 = PID 3 = VPU 4 = VPB | uint8 | 3101 | 12545 | Not applicable |
| AdvancedLoop.Setup.SlaveChannel2 | Slave cool/channel 2 control type (as above) | uint8 | 3102 | 12546 | Not applicable |
| AdvancedLoop.Setup.SlaveName | Slave loop name | string_t | 7020 | 28704 | Not applicable |
| AdvancedLoop.SlavePID.ActiveSet | Active set (1 = Set 1; 2 = Set 2; 3 = Set 3) | uint8 | 3138 | 12600 | Not applicable |
| AdvancedLoop.SlavePID.Boundary1-2 | Scheduler boundary 1-2 | float32 | 3139 | 12601 | 0dp |
| AdvancedLoop.SlavePID.Boundary2-3 | Scheduler boundary 2-3 | float32 | 133a | 4922 | 0dp |
| AdvancedLoop.SlavePID.ControlAction | Control action (0 = Reverse acting; 1 = Direct acting) | uint8 | 3106 | 12550 | Not applicable |
| AdvancedLoop.SlavePID.CutbackHigh | Cutback high set 1 (0 = Auto) | float32 | 313f | 12607 | 1dp |
| AdvancedLoop.SlavePID.CutbackHigh2 | Cutback high set 2 (0 = Auto) | float32 | 3147 | 12615 | 1dp |
| AdvancedLoop.SlavePID.CutbackHigh3 | Cutback high set 3 (0 = Auto) | float32 | 314f | 12623 | 1dp |
| AdvancedLoop.SlavePID.CutbackLow | Cutback low set 1 (0 = Auto) | float32 | 3140 | 12608 | 1dp |
| AdvancedLoop.SlavePID.CutbackLow2 | Cutback low set 2 (0 = Auto) | float32 | 3148 | 12616 | 1dp |
| AdvancedLoop.SlavePID.CutbackLow3 | Cutback low set 3 (0 = Auto) | float32 | 3150 | 12624 | 1dp |
| AdvancedLoop.SlavePID.DerivativeTime | Derivative time set 1 (0 = Off) | float32 | 313d | 12605 | 1dp |
| AdvancedLoop.SlavePID.DerivativeTime2 | Derivative time set 2 (0 = Off) | float32 | 3145 | 12613 | 1dp |
| AdvancedLoop.SlavePID.DerivativeTime3 | Derivative time set 3 (0 = Off) | float32 | 314d | 12621 | 1dp |
| AdvancedLoop.SlavePID.DerivativeType | Derivative type (0 = PV; 1 = Error) | uint8 | 3305 | 13061 | Not applicable |
| AdvancedLoop.SlavePID.IntegralTime | Integral time set 1 (0 = Off) | float32 | 313c | 12604 | 1dp |
| AdvancedLoop.SlavePID.IntegralTime2 | Integral time set 2 (0 = Off) | float32 | 3144 | 12612 | 1dp |
| AdvancedLoop.SlavePID.IntegralTime3 | Integral time set 3 (0 = Off) | float32 | 314c | 12620 | 1dp |
| AdvancedLoop.SlavePID.LoopBreakTime | Loop break time set 1 (0 = Off) | float32 | 3142 | 12610 | 0dp |
| AdvancedLoop.SlavePID.LoopBreakTime2 | Loop break time set 2 (0 = Off) | float32 | 314a | 12618 | 0dp |
| AdvancedLoop.SlavePID.LoopBreakTime3 | Loop break time set 3 (0 = Off) | float32 | 3152 | 12626 | 0dp |
| AdvancedLoop.SlavePID.ManualReset | Manual reset | float32 | 3141 | 12609 | 1dp |
| AdvancedLoop.SlavePID.ManualReset2 | Manual reset 2 | float32 | 3149 | 12617 | 1dp |
| AdvancedLoop.SlavePID.ManualReset3 | Manual reset 3 | float32 | 3151 | 12625 | 1dp |
| AdvancedLoop.SlavePID.NumberOfSets | Number of PID sets | uint8 | 3136 | 12598 | Not applicable |
| AdvancedLoop.SlavePID.OutputHi2 | Output high limit | float32 | 3155 | 12629 | 1dp |
| AdvancedLoop.SlavePID.OutputHi3 | Output high limit | float32 | 3157 | 12631 | 1dp |
| AdvancedLoop.SlavePID.OutputHigh | Output high limit | float32 | 3153 | 12627 | 1dp |
| AdvancedLoop.SlavePID.OutputLo2 | Output low limit 2 | float32 | 3156 | 12630 | 1dp |
| AdvancedLoop.SlavePID.OutputLo3 | Output low limit | float32 | 3158 | 12632 | 1dp |
| AdvancedLoop.SlavePID.OutputLow | Output low limit | float32 | 3154 | 12628 | 1dp |
| AdvancedLoop.SlavePID.PBUnits | Proportional band units (0 = Engineering; 1 = Percentage) | uint8 | 3304 | 13060 | Not applicable |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| AdvancedLoop.SlavePID.ProportionalBand | Proportional band set 1 | float32 | 313b | 12603 | 1dp |
| AdvancedLoop.SlavePID.ProportionalBand2 | Proportional band set 2 | float32 | 3143 | 12611 | 1dp |
| AdvancedLoop.SlavePID.ProportionalBand3 | Proportional band set 3 | float32 | 314b | 12619 | 1dp |
| AdvancedLoop.SlavePID.RelCh2Gain | Relative cool/channel 2 gain | float32 | 313e | 12606 | 1dp |
| AdvancedLoop.SlavePID.RelCh2Gain2 | Relative cool/channel 2 gain 2 | float32 | 3146 | 12614 | 1dp |
| AdvancedLoop.SlavePID.RelCh2Gain3 | Relative cool/channel 2 gain 3 | float32 | 314e | 12622 | 1dp |
| AdvancedLoop.SlavePID.RemoteInput | Scheduler remote input | float32 | 3137 | 12599 | 0dp |
| AdvancedLoop.SlavePID.SchedulerType | Scheduler type<br>0 = Off 1 = Manually set 2 = Setpoint 3 = PV<br>4 = Error 5 = Output 6 = Remote | uint8 | 3135 | 12597 | Not applicable |
| AdvancedLoop.SlaveSP.FFSelect | Feedforward select<br>0 = Master PV 1 = Master WSP 2 = Remote FF | uint8 | 31bf | 12735 | Not applicable |
| AdvancedLoop.SlaveSP.LocalSP | Local setpoint | float32 | 31b4 | 12724 | Same as AdvancedLoop.Main.SlavePV |
| AdvancedLoop.SlaveSP.ManualTrack | Manual track enable (0 = Off; 1 = On) | uint8 | 31ca | 12746 | Not applicable |
| AdvancedLoop.SlaveSP.MasterSensorBreakMode | Master sensor break mode<br>0 = SbrkSP 1 = Hold 2 = SlaveSB | uint8 | 31c2 | 12738 | Not applicable |
| AdvancedLoop.SlaveSP.RangeHigh | Range high | float32 | 31c0 | 12736 | Same as AdvancedLoop.Main.SlavePV |
| AdvancedLoop.SlaveSP.RangeLow | Range low | float32 | 31c1 | 12737 | Same as AdvancedLoop.Main.SlavePV |
| AdvancedLoop.SlaveSP.RemoteFeedForward | Remote feedforward input | float32 | 31bb | 12731 | Same as AdvancedLoop.Main.SlavePV |
| AdvancedLoop.SlaveSP.RemoteFFEnable | Remote feedforward enable (0 = No; 1 = Yes) | bool | 31be | 12734 | Not applicable |
| AdvancedLoop.SlaveSP.RemoteFFHigh | Remote feedforward high | float32 | 31bc | 12732 | Same as AdvancedLoop.Main.SlavePV |
| AdvancedLoop.SlaveSP.RemoteFFLow | Remote feddforward low | float32 | 31bd | 12733 | Same as AdvancedLoop.Main.SlavePV |
| AdvancedLoop.SlaveSP.SbrkSP | Sensor break setpoint | float32 | 31c3 | 12739 | Same as AdvancedLoop.Main.SlavePV |
| AdvancedLoop.SlaveSP.SPHighLimit | Setpoint high limit | float32 | 31b5 | 12725 | Same as AdvancedLoop.Main.SlavePV |
| AdvancedLoop.SlaveSP.SPLowLimit | Setpoint low limit | float32 | 31b6 | 12726 | Same as AdvancedLoop.Main.SlavePV |
| AdvancedLoop.SlaveSP.TrimHighLimit | Trim high limit | float32 | 31b9 | 12729 | Same as AdvancedLoop.Main.MasterPV |
| AdvancedLoop.SlaveSP.TrimLowLimit | Trim low limit | float32 | 31ba | 12730 | Same as AdvancedLoop.Main.MasterPV |
| AdvancedLoop.SlaveSP.TrimRangeHigh | Trim range high | float32 | 31b7 | 12727 | Same as AdvancedLoop.Main.SlavePV |
| AdvancedLoop.SlaveSP.TrimRangeLow | Trim range low | float32 | 31b8 | 12728 | Same as AdvancedLoop.Main.SlavePV |
| AdvancedLoop.Tune.A1 | A1 | float32 | 320d | 12813 | 0dp |
| AdvancedLoop.Tune.A2 | A2 | float32 | 320e | 12814 | 0dp |
| AdvancedLoop.Tune.Alpha | Alpha | float32 | 3211 | 12817 | 4dp |
| AdvancedLoop.Tune.Alpha_p | Alpha_p | float32 | 320f | 12815 | 2dp |
| AdvancedLoop.Tune.ArgOP | Argument Output | float32 | 3209 | 12809 | 1dp |
| AdvancedLoop.Tune.ArgPV | Argument PV | float32 | 3208 | 12808 | 1dp |
| AdvancedLoop.Tune.Band | Band | float32 | 31c7 | 12743 | 1dp |
| AdvancedLoop.Tune.CycleNo | CycleNo | float32 | 3213 | 12819 | 0dp |
| AdvancedLoop.Tune.Debug | Debug | float32 | 3212 | 12818 | 2dp |
| AdvancedLoop.Tune.Diagnostics | Tuning diagnostics | bool | 31cb | 12747 | Not applicable |
| AdvancedLoop.Tune.Gain | Gain | float32 | 320a | 12810 | 1dp |
| AdvancedLoop.Tune.Hysteresis | Hysteresis | float32 | 31c6 | 12742 | 1dp |
| AdvancedLoop.Tune.MasterTune | Master tune | float32 | 3203 | 12803 | 0dp |
| AdvancedLoop.Tune.ModeMan | Mode Man | float32 | 3201 | 12801 | 0dp |
| AdvancedLoop.Tune.ModOP | Modulus OP | float32 | 3207 | 12807 | 1dp |
| AdvancedLoop.Tune.ModPV | Modulus PV | float32 | 3206 | 12806 | 1dp |
| AdvancedLoop.Tune.OP | Output | float32 | 3202 | 12802 | 1dp |
| AdvancedLoop.Tune.OPDel | OPDel | float32 | 0319 | 793 | 2dp |
| AdvancedLoop.Tune.OPss | OPss | float32 | 3210 | 12816 | 2dp |
| AdvancedLoop.Tune.OutputHighLimit | Output high | float32 | 3132 | 12594 | Same as AdvancedLoop.Output.Out-putHighLimit |
| AdvancedLoop.Tune.OutputLowLimit | Output low | float32 | 3133 | 12595 | Same as AdvancedLoop.Output.Out-putHighLimit |
| AdvancedLoop.Tune.PBs | PBs | float32 | 3214 | 12820 | 2dp |
| AdvancedLoop.Tune.Period | Period | float32 | 320c | 12812 | 0dp |
| AdvancedLoop.Tune.Phase | Phase | float32 | 320b | 12811 | 1dp |
| AdvancedLoop.Tune.Settle | Settle | float32 | 3216 | 12822 | 2dp |
| AdvancedLoop.Tune.Stage | Stage<br>0 = Reset 1 = None 2 = Settling 3 = Current SP<br>4 = New PP 5 = To SP 6 = Wait Max. 7 = Wait Min<br>8 = Store 9 = CoolT 10 = PID 11 = Abort<br>12 = Complete 13 = New R2g 14 = 1: Half Cycle<br>15 = 2: Full Cycle 16 = 3: Full Cycle<br>17 = 4: Final cycle 18 = 5: Calculating | uint8 | 0308 | 776 | Not applicable |
| AdvancedLoop.Tune.StageTime | Stage time | float32 | 0309 | 777 | 0dp |
| AdvancedLoop.Tune.State | State<br>0 = Off 1 = Ready 2 - Running 3 = Complete<br>4 = Time-out 5 = Ti Limit 6 = R2G limit | uint8 | 0307 | 775 | Not applicable |
| AdvancedLoop.Tune.TDs | TDs | float32 | 3215 | 12821 | 2dp |
| AdvancedLoop.Tune.Timeout | Timeout | float32 | 0326 | 806 | 0dp |
| AdvancedLoop.Tune.TuneEnable | Autotune enable (0 = Off; 1 = On) | bool | 3131 | 12593 | Not applicable |
| AdvancedLoop.Tune.TuneHigh | Tune high | float32 | 31c8 | 12744 | Same as AdvancedLoop.Main.SlavePV |
| AdvancedLoop.Tune.TuneLow | Tune low | float32 | 31c9 | 12745 | Same as AdvancedLoop.Main.SlavePV |
| AdvancedLoop.Tune.TuneR2G | Slave R2G tuning type<br>0 = Standard R2G tuning 1 = R2GPD tuning 2 = Off | uint8 | 3130 | 12592 | Not applicable |
| AdvancedLoop.Tune.TuneSlave | Tune slave | float32 | 3204 | 12804 | 1dp |
| AdvancedLoop.Tune.TuneStatus | Tune Status<br>0 = Not tuning 1 = Tuning the slave<br>2 = Tuning the master 3 = Tuning complete<br>-1 = Tuning aborted or timed-out | float32 | 3205 | 12805 | 0dp |
| AdvancedLoop.Tune.TuneType | Autotune algorithm type (0 = Slave; 1 = Master) | uint8 | 31c5 | 12741 | Not applicable |
| AdvancedLoop.Tune.WSP | Working setpoint | float32 | 3200 | 12800 | Same as AdvancedLoop.Main.MasterPV |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| AlarmSummary.AnyAlarm | 0 = No active alarms; 1 = one or more alarms active | bool | 01a2 | 418 | Not applicable |
| AlarmSummary.AnyChanAlarm | 0 = No channel alarms<br>1 = Channel alarm(s) active but all ack'd.<br>2 = Channel alarm(s) active but not all ack'd | uint8 | 01a0 | 416 | Not applicable |
| AlarmSummary.AnySystemAlarm | 0 = No system alarms; 1 = 1 or more system alm(s) | bool | 01a1 | 417 | Not applicable |
| AlarmSummary.Channel.Alarm1Ack | Acknowledge the most recent channel alarm | bool | 1192 | 4498 | Not applicable |
| AlarmSummary.Channel.Alarm1Num | Channel and alarm number of most recent alarm<br>0 = No alarm 4 = Ch1;Al1 5 =Ch1;Al2<br>8 = Ch2;Al1 9 = Ch2Al2 12 = Ch3;Al1<br>13 = Ch3;Al2 16 = Ch4;Al1 17 = Ch4;Al2<br>132 = VC1;Al1 133 = VC1;Al2 136 = VC2;Al1<br>137 = VC2;Al2 140 = VC3;Al1 141 = VC3;Al2<br>144 = VC4;Al1 145 = VC4;Al2 148 = VC5;Al1<br>149 = VC5;Al2 152 = VC6;Al1 153 = VC6;Al2<br>156 = VC7;Al1 157 = VC7;Al2 160 = VC8;Al1<br>161 = VC8;Al2 164 = VC9;Al1 165 = VC9;Al2<br>168 = VC10;Al1 169 = VC10;Al2 172 = VC11;Al1<br>173 = VC11;Al2 176 = VC12;Al1 177 = VC12;Al2<br>180 = VC13;Al1 181 = VC13;Al2 184 = VC14;Al1<br>185 = VC14;Al2 188 = VC15;Al1 189 = VC15:Al2 | uint8 | 1190 | 4496 | Not applicable |
| AlarmSummary.Channel.Alarm1Status | Status of most recent alarm<br>0 = Off 1 = Active 2 = Safe unack 3 = Active unack | uint8 | 1191 | 4497 | Not applicable |
| AlarmSummary.Channel.Alarm2Ack | Acknowledge the 2nd most recent channel alarm | bool | 1195 | 4501 | Not applicable |
| AlarmSummary.Channel.Alarm2Num | As Alarm1Num, but for 2nd most recent alarm | uint8 | 1193 | 4499 | Not applicable |
| AlarmSummary.Channel.Alarm2Status | As Alarm1Status, but for 2nd most recent alarm | uint8 | 1194 | 4500 | Not applicable |
| AlarmSummary.Channel.Alarm3Ack | Acknowledge the 3rd most recent channel alarm | bool | 1198 | 4504 | Not applicable |
| AlarmSummary.Channel.Alarm3Num | As Alarm1Num, but for 3rd most recent alarm | uint8 | 1196 | 4502 | Not applicable |
| AlarmSummary.Channel.Alarm3Status | As Alarm1Status, but for 3rd most recent alarm | uint8 | 1197 | 4503 | Not applicable |
| AlarmSummary.Channel.Alarm4Ack | Acknowledge the 4th most recent channel alarm | bool | 119b | 4507 | Not applicable |
| AlarmSummary.Channel.Alarm4Num | As Alarm1Num, but for 4th most recent alarm | uint8 | 1199 | 4505 | Not applicable |
| AlarmSummary.Channel.Alarm4Status | As Alarm1Status, but for 4th most recent alarm | uint8 | 119a | 4506 | Not applicable |
| AlarmSummary.Channel.Alarm5Ack | Acknowledge the 5th most recent channel alarm | bool | 119e | 4510 | Not applicable |
| AlarmSummary.Channel.Alarm5Num | As Alarm1Num, but for 5th most recent alarm | uint8 | 119c | 4508 | Not applicable |
| AlarmSummary.Channel.Alarm5Status | As Alarm1Status, but for 5th most recent alarm | uint8 | 119d | 4509 | Not applicable |
| AlarmSummary.Channel.Alarm6Ack | Acknowledge the 6th most recent channel alarm | bool | 11a1 | 4513 | Not applicable |
| AlarmSummary.Channel.Alarm6Num | As Alarm1Num, but for 6th most recent alarm | uint8 | 119f | 4511 | Not applicable |
| AlarmSummary.Channel.Alarm6Status | As Alarm1Status, but for 6th most recent alarm | uint8 | 11a0 | 4512 | Not applicable |
| AlarmSummary.Channel.Alarm7Ack | Acknowledge the 7th most recent channel alarm | bool | 11a4 | 4516 | Not applicable |
| AlarmSummary.Channel.Alarm7Num | As Alarm1Num, but for 7th most recent alarm | uint8 | 11a2 | 4514 | Not applicable |
| AlarmSummary.Channel.Alarm7Status | As Alarm1Status, but for 7th most recent alarm | uint8 | 11a3 | 4515 | Not applicable |
| AlarmSummary.Channel.Alarm8Ack | Acknowledge the 8th most recent channel alarm | bool | 11a7 | 4519 | Not applicable |
| AlarmSummary.Channel.Alarm8Num | As Alarm1Num, but for 8th most recent alarm | uint8 | 11a5 | 4517 | Not applicable |
| AlarmSummary.Channel.Alarm8Status | As Alarm1Status, but for 8th most recent alarm | uint8 | 11a6 | 4518 | Not applicable |
| AlarmSummary.Channel.Alarm9Ack | Acknowledge the 9th most recent channel alarm | bool | 11aa | 4522 | Not applicable |
| AlarmSummary.Channel.Alarm9Num | As Alarm1Num, but for 9th most recent alarm | uint8 | 11a8 | 4520 | Not applicable |
| AlarmSummary.Channel.Alarm9Status | As Alarm1Status, but for 9th most recent alarm | uint8 | 11a9 | 4521 | Not applicable |
| AlarmSummary.Channel.Alarm10Ack | Acknowledge the 10th most recent channel alarm | bool | 11ad | 4525 | Not applicable |
| AlarmSummary.Channel.Alarm10Num | As Alarm1Num, but for 10th most recent alarm | uint8 | 11ab | 4523 | Not applicable |
| AlarmSummary.Channel.Alarm10Status | As Alarm1Status, but for 10th most recent alarm | uint8 | 11ac | 4524 | Not applicable |
| AlarmSummary.Channel.Alarm11Ack | Acknowledge the 11th most recent channel alarm | bool | 11b0 | 4528 | Not applicable |
| AlarmSummary.Channel.Alarm11Num | As Alarm1Num, but for 11th most recent alarm | uint8 | 11ae | 4526 | Not applicable |
| AlarmSummary.Channel.Alarm11Status | As Alarm1Status, but for 11th most recent alarm | uint8 | 11af | 4527 | Not applicable |
| AlarmSummary.Channel.Alarm12Ack | Acknowledge the 12th most recent channel alarm | bool | 11b3 | 4531 | Not applicable |
| AlarmSummary.Channel.Alarm12Num | As Alarm1Num, but for 12th most recent alarm | uint8 | 11b1 | 4529 | Not applicable |
| AlarmSummary.Channel.Alarm12Status | As Alarm1Status, but for 12th most recent alarm | uint8 | 11b2 | 4530 | Not applicable |
| AlarmSummary.Channel.Alarm13Ack | Acknowledge the 13th most recent channel alarm | bool | 11b6 | 4534 | Not applicable |
| AlarmSummary.Channel.Alarm13Num | As Alarm1Num, but for 13th most recent alarmr | uint8 | 11b4 | 4532 | Not applicable |
| AlarmSummary.Channel.Alarm13Status | As Alarm1Status, but for 13th most recent alarm | uint8 | 11b5 | 4533 | Not applicable |
| AlarmSummary.Channel.Alarm14Ack | Acknowledge the 14th most recent channel alarm | bool | 11b9 | 4537 | Not applicable |
| AlarmSummary.Channel.Alarm14Num | As Alarm1Num, but for 14th most recent alarmr | uint8 | 11b7 | 4535 | Not applicable |
| AlarmSummary.Channel.Alarm14Status | As Alarm1Status, but for 14th most recent alarm | uint8 | 11b8 | 4536 | Not applicable |
| AlarmSummary.Channel.Alarm15Ack | Acknowledge the 15th most recent channel alarm | bool | 11bc | 4540 | Not applicable |
| AlarmSummary.Channel.Alarm15Num | As Alarm1Num, but for 15th most recent alarm | uint8 | 11ba | 4538 | Not applicable |
| AlarmSummary.Channel.Alarm15Status | As Alarm1Status, but for 15th most recent alarm | uint8 | 11bb | 4539 | Not applicable |
| AlarmSummary.Channel.Alarm16Ack | Acknowledge the 16th most recent channel alarm | bool | 11bf | 4543 | Not applicable |
| AlarmSummary.Channel.Alarm16Num | As Alarm1Num, but for 16th most recent alarm | uint8 | 11bd | 4541 | Not applicable |
| AlarmSummary.Channel.Alarm16Status | As Alarm1Status, but for 16th most recent alarm | uint8 | 11be | 4542 | Not applicable |
| AlarmSummary.Channel.Alarm17Ack | Acknowledge the 17th most recent channel alarm | bool | 11c2 | 4546 | Not applicable |
| AlarmSummary.Channel.Alarm17Num | As Alarm1Num, but for 17th most recent alarm | uint8 | 11c0 | 4544 | Not applicable |
| AlarmSummary.Channel.Alarm17Status | As Alarm1Status, but for 17th most recent alarm | uint8 | 11c1 | 4545 | Not applicable |
| AlarmSummary.Channel.Alarm18Ack | Acknowledge the 18th most recent channel alarm | bool | 11c5 | 4549 | Not applicable |
| AlarmSummary.Channel.Alarm18Num | As Alarm1Num, but for 18th most recent alarm | uint8 | 11c3 | 4547 | Not applicable |
| AlarmSummary.Channel.Alarm18Status | As Alarm1Status, but for 18th most recent alarm | uint8 | 11c4 | 4548 | Not applicable |
| AlarmSummary.Channel.Alarm19Ack | Acknowledge the 19th most recent channel alarm | bool | 11c8 | 4552 | Not applicable |
| AlarmSummary.Channel.Alarm19Num | As Alarm1Num, but for 19th most recent alarm | uint8 | 11c6 | 4550 | Not applicable |
| AlarmSummary.Channel.Alarm19Status | As Alarm1Status, but for 19th most recent alarm | uint8 | 11c7 | 4551 | Not applicable |
| AlarmSummary.Channel.Alarm20Ack | Acknowledge the 20th most recent channel alarm | bool | 11cb | 4555 | Not applicable |
| AlarmSummary.Channel.Alarm20Num | As Alarm1Num, but for 20th most recent alarm | uint8 | 11c9 | 4553 | Not applicable |
| AlarmSummary.Channel.Alarm20Status | As Alarm1Status, but for 20th most recent alarm | uint8 | 11ca | 4554 | Not applicable |
| AlarmSummary.Channel.Alarm21Ack | Acknowledge the 21st most recent channel alarm | bool | 11ce | 4558 | Not applicable |
| AlarmSummary.Channel.Alarm21Num | As Alarm1Num, but for 21st most recent alarm | uint8 | 11cc | 4556 | Not applicable |
| AlarmSummary.Channel.Alarm21Status | As Alarm1Status, but for 21st most recent alarm | uint8 | 11cd | 4557 | Not applicable |
| AlarmSummary.Channel.Alarm22Ack | Acknowledge the 22nd most recent channel alarm | bool | 11d1 | 4561 | Not applicable |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| AlarmSummary.Channel.Alarm22Num | As Alarm1Num, but for 22nd most recent alarm | uint8 | 11cf | 4559 | Not applicable |
| AlarmSummary.Channel.Alarm22Status | As Alarm1Status, but for 22nd most recent alarm | uint8 | 11d0 | 4560 | Not applicable |
| AlarmSummary.Channel.Alarm23Ack | Acknowledge the 23rd most recent channel alarm | bool | 11d4 | 4564 | Not applicable |
| AlarmSummary.Channel.Alarm23Num | As Alarm1Num, but for 23th most recent alarm | uint8 | 11d2 | 4562 | Not applicable |
| AlarmSummary.Channel.Alarm23Status | As Alarm1Status, but for 23rd most recent alarm | uint8 | 11d3 | 4563 | Not applicable |
| AlarmSummary.Channel.Alarm24Ack | Acknowledge the 24th most recent channel alarm | bool | 11d7 | 4567 | Not applicable |
| AlarmSummary.Channel.Alarm24Num | As Alarm1Num, but for 24th most recent alarm | uint8 | 11d5 | 4565 | Not applicable |
| AlarmSummary.Channel.Alarm24Status | As Alarm1Status, but for 24th most recent alarm | uint8 | 11d6 | 4566 | Not applicable |
| AlarmSummary.Channel.Alarm25Ack | Acknowledge the 25th most recent channel alarm | bool | 11da | 4570 | Not applicable |
| AlarmSummary.Channel.Alarm25Num | As Alarm1Num, but for 25th most recent alarm | uint8 | 11d8 | 4568 | Not applicable |
| AlarmSummary.Channel.Alarm25Status | As Alarm1Status, but for 25th most recent alarm | uint8 | 11d9 | 4569 | Not applicable |
| AlarmSummary.Channel.Alarm26Ack | Acknowledge the 26th most recent channel alarm | bool | 11dd | 4573 | Not applicable |
| AlarmSummary.Channel.Alarm26Num | As Alarm1Num, but for 26th most recent alarm | uint8 | 11db | 4571 | Not applicable |
| AlarmSummary.Channel.Alarm26Status | As Alarm1Status, but for 26th most recent alarm | uint8 | 11dc | 4572 | Not applicable |
| AlarmSummary.Channel.Alarm27Ack | Acknowledge the 27th most recent channel alarm | bool | 11e0 | 4576 | Not applicable |
| AlarmSummary.Channel.Alarm27Num | As Alarm1Num, but for 27th most recent alarm | uint8 | 11de | 4574 | Not applicable |
| AlarmSummary.Channel.Alarm27Status | As Alarm1Status, but for 27th most recent alarm | uint8 | 11df | 4575 | Not applicable |
| AlarmSummary.Channel.Alarm28Ack | Acknowledge the 28th most recent channel alarm | bool | 11e3 | 4579 | Not applicable |
| AlarmSummary.Channel.Alarm28Num | As Alarm1Num, but for 28th most recent alarm | uint8 | 11e1 | 4577 | Not applicable |
| AlarmSummary.Channel.Alarm28Status | As Alarm1Status, but for 28th most recent alarm | uint8 | 11e2 | 4578 | Not applicable |
| AlarmSummary.Channel.Alarm29Ack | Acknowledge the 29th most recent channel alarm | bool | 11e6 | 4582 | Not applicable |
| AlarmSummary.Channel.Alarm29Num | As Alarm1Num, but for 29th most recent alarm | uint8 | 11e4 | 4580 | Not applicable |
| AlarmSummary.Channel.Alarm29Status | As Alarm1Status, but for 29th most recent alarm | uint8 | 11e5 | 4581 | Not applicable |
| AlarmSummary.Channel.Alarm30Ack | Acknowledge the 30th most recent channel alarm | bool | 11e9 | 4585 | Not applicable |
| AlarmSummary.Channel.Alarm30Num | As Alarm1Num, but for 30th most recent alarm | uint8 | 11e7 | 4583 | Not applicable |
| AlarmSummary.Channel.Alarm30Status | As Alarm1Status, but for 30th most recent alarm | uint8 | 11e8 | 4584 | Not applicable |
| AlarmSummary.Channel.Alarm31Ack | Acknowledge the 31st most recent channel alarm | bool | 11ec | 4588 | Not applicable |
| AlarmSummary.Channel.Alarm31Num | As Alarm1Num, but for 31st most recent alarm | uint8 | 11ea | 4586 | Not applicable |
| AlarmSummary.Channel.Alarm31Status | As Alarm1Status, but for 31st most recent alarm | uint8 | 11eb | 4587 | Not applicable |
| AlarmSummary.Channel.Alarm32Ack | Acknowledge the 32nd most recent channel alarm | bool | 11ef | 4591 | Not applicable |
| AlarmSummary.Channel.Alarm32Num | As Alarm1Num, but for 32nd most recent alarm | uint8 | 11ed | 4589 | Not applicable |
| AlarmSummary.Channel.Alarm32Status | As Alarm1Status, but for 32nd most recent alarm | uint8 | 11ee | 4590 | Not applicable |
| AlarmSummary.Channel.Alarm33Ack | Acknowledge the 33rd most recent channel alarm | bool | 11f2 | 4594 | Not applicable |
| AlarmSummary.Channel.Alarm33Num | As Alarm1Num, but for 33rd most recent alarm | uint8 | 11f0 | 4592 | Not applicable |
| AlarmSummary.Channel.Alarm33Status | As Alarm1Status, but for 33rd most recent alarm | uint8 | 11f1 | 4593 | Not applicable |
| AlarmSummary.Channel.Alarm34Ack | Acknowledge the 34th most recent channel alarm | bool | 11f5 | 4597 | Not applicable |
| AlarmSummary.Channel.Alarm34Num | As Alarm1Num, but for 34th most recent alarm | uint8 | 11f3 | 4595 | Not applicable |
| AlarmSummary.Channel.Alarm34Status | As Alarm1Status, but for 34th most recent alarm | uint8 | 11f4 | 4596 | Not applicable |
| AlarmSummary.Channel.Alarm35Ack | Acknowledge the 35th most recent channel alarm | bool | 11f8 | 4600 | Not applicable |
| AlarmSummary.Channel.Alarm35Num | As Alarm1Num, but for 35th most recent alarm | uint8 | 11f6 | 4598 | Not applicable |
| AlarmSummary.Channel.Alarm35Status | As Alarm1Status, but for 35th most recent alarm | uint8 | 11f7 | 4599 | Not applicable |
| AlarmSummary.Channel.Alarm36Ack | Acknowledge the 36th most recent channel alarm | bool | 11fb | 4603 | Not applicable |
| AlarmSummary.Channel.Alarm36Num | As Alarm1Num, but for 36th most recent alarm | uint8 | 11f9 | 4601 | Not applicable |
| AlarmSummary.Channel.Alarm36Status | As Alarm1Status, but for 36th most recent alarm | uint8 | 11fa | 4602 | Not applicable |
| AlarmSummary.Channel.Alarm37Ack | Acknowledge the 37th most recent channel alarm | bool | 11fe | 4606 | Not applicable |
| AlarmSummary.Channel.Alarm37Num | As Alarm1Num, but for 37th most recent alarm | uint8 | 11fc | 4604 | Not applicable |
| AlarmSummary.Channel.Alarm37Status | As Alarm1Status, but for 38th most recent alarm | uint8 | 11fd | 4605 | Not applicable |
| AlarmSummary.Channel.Alarm38Ack | Acknowledge the 38th most recent channel alarm | bool | 1201 | 4609 | Not applicable |
| AlarmSummary.Channel.Alarm38Num | As Alarm1Num, but for 38th most recent alarm | uint8 | 11ff | 4607 | Not applicable |
| AlarmSummary.Channel.Alarm38Status | As Alarm1Status, but for 38th most recent alarm | uint8 | 1200 | 4608 | Not applicable |
| AlarmSummary.GlobalAck | Acknowledge all alarms. 0=No;1 = yes | bool | 01a3 | 419 | Not applicable |
| AlarmSummary.StatusWord1 | A summary of Channel 1-4 alarms<br>Bit 0: 1 = Channel 1 Alarm 1 active<br>Bit 1: 1 = Channel 1 Alarm 1 not acknowledged<br>Bit 2: 1 = Channel 1 Alarm 2 active<br>Bit 3: 1 = Channel 1 Alarm 2 not acknowledged<br>Bit 4: 1 = Channel 2 Alarm 1 active<br>Bit 5: 1 = Channel 2 Alarm 1 not acknowledged<br>Bit 6: 1 = Channel 2 Alarm 2 active<br>Bit 7: 1 = Channel 2 Alarm 2 not acknowledged<br>Bit 8: 1 = Channel 3 Alarm 1 active<br>Bit 9: 1 = Channel 3 Alarm 1 not acknowledged<br>Bit 10: 1 = Channel 3 Alarm 2 active<br>Bit 11: 1 = Channel 3 Alarm 2 not acknowledged<br>Bit 12: 1 = Channel 4 Alarm 1 active<br>Bit 13: 1 = Channel 4 Alarm 1 not acknowledged<br>Bit 14: 1 = Channel 4 Alarm 2 active<br>Bit 15: 1 = Channel 4 Alarm 2 not acknowledged | int16 | 01a4 | 420 | Not applicable |
| AlarmSummary.StatusWord2 | A summary of Virtual Channel 1 to 4 alarms<br>Bit 0: 1 = Virtual channel 1 Alarm 1 active<br>Bit 1: 1 = Virtual channel 1 Alarm 1 not ack'd<br>Bit 2: 1 = Virtual channel 1 Alarm 2 active<br>Bit 3: 1 = Virtual channel 1 Alarm 2 not ack'd<br>Bit 4: 1 = Virtual channel 2 Alarm 1 active<br>Bit 5: 1 = Virtual channel 2 Alarm 1 not ack'd<br>Bit 6: 1 = Virtual channel 2 Alarm 2 active<br>Bit 7: 1 = Virtual channel 2 Alarm 2 not ack'd<br>Bit 8: 1 = Virtual channel 3 Alarm 1 active<br>Bit 9: 1 = Virtual channel 3 Alarm 1 not ack'd<br>Bit 10: 1 = Virtual channel 3 Alarm 2 active<br>Bit 11: 1 = Virtual channel 3 Alarm 2 not ack'd | int16 | 01a5 | 421 | Not applicable |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| AlarmSummary.StatusWord2 (Cont.) | Bit 12: 1 = Virtual channel 4 Alarm 1 active<br>Bit 13: 1 = Virtual channel 4 Alarm 1 not ack'd<br>Bit 14: 1 = Virtual channel 4 Alarm 2 active<br>Bit 15: 1 = Virtual channel 4 Alarm 2 not ack'd | | | | |
| AlarmSummary.StatusWord3 | A summary of Virtual Channel 5 to 8 alarms<br>As for Status Word 2 but for virtual channs 5 to 8 | int16 | 01a6 | 422 | Not applicable |
| AlarmSummary.StatusWord4 | A summary of Virtual Channel 9 to 12 alarms<br>As for Status Word 2 but for virtual channs 9 to 12 | int16 | 01a7 | 423 | Not applicable |
| AlarmSummary.StatusWord5 | A summary of Virtual Channel 13 to 14 alarms<br>As for Status Word 2 but for virtual channs 13 to 15 | int16 | 01a8 | 424 | Not applicable |
| AlarmSummary.System.Alarm1ID | Most recent active system alarm<br>0 = No Alarm 1 = Low battery<br>2 = Battery failure 3 = System clock fail<br>4 = Channel error 5 = Channel fail<br>6 = DHCP server fail 7 = FTP Archive file lost<br>8 = FTP Archive slow 9 = FTP Primary server failure<br>10 = FTP Secondary server failure<br>11 = Insufficient non-volatile memory<br>12 = Maths channel failure 13 = Media archive file lost<br>14 = Media archive slow 15 = Network boot failure<br>16 = DC Output Cal. Error 17 = Recording failure<br>18 = Media failure 19: = Media full<br>20 =SNTP failure 21 = Time synchronisation failure<br>22 = Media missing 23: Archive disabled<br>24 = Archiving failed 25 = Archiving timed out<br>26 = USB Over Current 27 = USB unsuported<br>28 = Invalid parameter database<br>29 = Invalid non-volatile data<br>30 = Flash write failure 31 = Wiring failure<br>32 = Broadcast Storm<br>33 = Non-volatile memory write frequency warning | uint8 | 1210 | 4624 | Not applicable |
| AlarmSummary.System.Alarm2ID | 2nd most recent active system alarm (as Alarm1ID) | uint8 | 1211 | 4625 | Not applicable |
| AlarmSummary.System.Alarm3ID | 3rd most recent active system alarm (as Alarm1ID) | uint8 | 1212 | 4626 | Not applicable |
| AlarmSummary.System.Alarm4ID | 4th most recent active system alarm (as Alarm1ID) | uint8 | 1213 | 4627 | Not applicable |
| AlarmSummary.System.Alarm5ID | 5th most recent active system alarm (as Alarm1ID) | uint8 | 1214 | 4628 | Not applicable |
| AlarmSummary.System.Alarm6ID | 6th most recent active system alarm (as Alarm1ID) | uint8 | 1215 | 4629 | Not applicable |
| AlarmSummary.System.Alarm7ID | 7th most recent active system alarm (as Alarm1ID) | uint8 | 1216 | 4630 | Not applicable |
| AlarmSummary.System.Alarm8ID | 8th most recent active system alarm (as Alarm1ID) | uint8 | 1217 | 4631 | Not applicable |
| AlarmSummary.System.Alarm9ID | 9th most recent active system alarm (as Alarm1ID) | uint8 | 1218 | 4632 | Not applicable |
| AlarmSummary.System.Alarm10ID | 10th most recent active system alarm (as Alarm1ID) | uint8 | 1219 | 4633 | Not applicable |
| AlarmSummary.System.Alarm11ID | 11th most recent active system alarm (as Alarm1ID) | uint8 | 121a | 4634 | Not applicable |
| AlarmSummary.System.Alarm12ID | 12th most recent active system alarm (as Alarm1ID) | uint8 | 121b | 4635 | Not applicable |
| AlarmSummary.System.Alarm13ID | 13th most recent active system alarm (as Alarm1ID) | uint8 | 121c | 4636 | Not applicable |
| AlarmSummary.System.Alarm14ID | 14th most recent active system alarm (as Alarm1ID) | uint8 | 121d | 4637 | Not applicable |
| AlarmSummary.System.Alarm15ID | 15th most recent active system alarm (as Alarm1ID) | uint8 | 121e | 4638 | Not applicable |
| AlarmSummary.System.Alarm16ID | 16th most recent active system alarm (as Alarm1ID) | uint8 | 121f | 4639 | Not applicable |
| AlarmSummary.System.Alarm17ID | 17th most recent active system alarm (as Alarm1ID) | uint8 | 1220 | 4640 | Not applicable |
| AlarmSummary.System.Alarm18ID | 18th most recent active system alarm (as Alarm1ID) | uint8 | 1221 | 4641 | Not applicable |
| AlarmSummary.System.Alarm19ID | 19th most recent active system alarm (as Alarm1ID) | uint8 | 1222 | 4642 | Not applicable |
| AlarmSummary.System.Alarm20ID | 20th most recent active system alarm (as Alarm1ID) | uint8 | 1223 | 4643 | Not applicable |
| AlarmSummary.System.Alarm21ID | 21st most recent active system alarm (as Alarm1ID) | uint8 | 1224 | 4644 | Not applicable |
| AlarmSummary.System.Alarm22ID | 22nd most recent active system alarm (as Alarm1ID) | uint8 | 1225 | 4645 | Not applicable |
| AlarmSummary.System.Alarm23ID | 23rd most recent active system alarm (as Alarm1ID) | uint8 | 1226 | 4646 | Not applicable |
| AlarmSummary.System.Alarm24ID | 24th most recent active system alarm (as Alarm1ID) | uint8 | 1227 | 4647 | Not applicable |
| AlarmSummary.System.Alarm25ID | 25th most recent active system alarm (as Alarm1ID) | uint8 | 1228 | 4648 | Not applicable |
| AlarmSummary.System.Alarm26ID | 26th most recent active system alarm (as Alarm1ID) | uint8 | 1229 | 4649 | Not applicable |
| AlarmSummary.System.Alarm27ID | 27th most recent active system alarm (as Alarm1ID) | uint8 | 122a | 4650 | Not applicable |
| AlarmSummary.System.Alarm28ID | 28th most recent active system alarm (as Alarm1ID) | uint8 | 122b | 4651 | Not applicable |
| AlarmSummary.System.Alarm29ID | 29th most recent active system alarm (as Alarm1ID) | uint8 | 122c | 4652 | Not applicable |
| AlarmSummary.System.Alarm30ID | 30th most recent active system alarm (as Alarm1ID) | uint8 | 122d | 4653 | Not applicable |
| AlarmSummary.System.Alarm31ID | 31st most recent active system alarm (as Alarm1ID) | uint8 | 122e | 4654 | Not applicable |
| AlarmSummary.System.Alarm32ID | 32nd most recent active system alarm (as Alarm1ID) | uint8 | 122f | 4655 | Not applicable |
| BCDInput.1.BCDVal | BCD1 BCD Value | uint8 | 2ed1 | 11985 | Not applicable |
| BCDInput.1.DecByte | BCD1 Decimal Value | uint8 | 2ed0 | 11984 | Not applicable |
| BCDInput.1.In1 | BCD1 Input 1 (0 = Off; 1 = On) | bool | 2ec8 | 11976 | Not applicable |
| BCDInput.1.In2 | BCD1 Input 2 (0 = Off; 1 = On) | bool | 2ec9 | 11977 | Not applicable |
| BCDInput.1.In3 | BCD1 Input 3 (0 = Off; 1 = On) | bool | 2eca | 11978 | Not applicable |
| BCDInput.1.In4 | BCD1 Input 4 (0 = Off; 1 = On) | bool | 2ecb | 11979 | Not applicable |
| BCDInput.1.In5 | BCD1 Input 5 (0 = Off; 1 = On) | bool | 2ecc | 11980 | Not applicable |
| BCDInput.1.In6 | BCD1 Input 6 (0 = Off; 1 = On) | bool | 2ecd | 11981 | Not applicable |
| BCDInput.1.In7 | BCD1 Input 7 (0 = Off; 1 = On) | bool | 2ece | 11982 | Not applicable |
| BCDInput.1.In8 | BCD1 Input 8 (0 = Off; 1 = On) | bool | 2ecf | 11983 | Not applicable |
| BCDInput.1.Tens | BCD1 Tens (MSD) | uint8 | 2ed3 | 11987 | Not applicable |
| BCDInput.1.Units | BCD1 Units (LSD) | uint8 | 2ed2 | 11986 | Not applicable |
| BCDInput.2.BCDVal | BCD2 BCD Value | uint8 | 2edd | 11997 | Not applicable |
| BCDInput.2.DecByte | BCD2 Decimal Value | uint8 | 2edc | 11996 | Not applicable |
| BCDInput.2.In1 | BCD2 Input 1 (0 = Off; 1 = On) | bool | 2ed4 | 11988 | Not applicable |
| BCDInput.2.In2 | BCD2 Input 2 (0 = Off; 1 = On) | bool | 2ed5 | 11989 | Not applicable |
| BCDInput.2.In3 | BCD2 Input 3 (0 = Off; 1 = On) | bool | 2ed6 | 11990 | Not applicable |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| BCDInput.2.In4 | BCD2 Input 4 (0 = Off; 1 = On) | bool | 2ed7 | 11991 | Not applicable |
| BCDInput.2.In5 | BCD2 Input 5 (0 = Off; 1 = On) | bool | 2ed8 | 11992 | Not applicable |
| BCDInput.2.In6 | BCD2 Input 6 (0 = Off; 1 = On) | bool | 2ed9 | 11993 | Not applicable |
| BCDInput.2.In7 | BCD2 Input 7 (0 = Off; 1 = On) | bool | 2eda | 11994 | Not applicable |
| BCDInput.2.In8 | BCD2 Input 8 (0 = Off; 1 = On) | bool | 2edb | 11995 | Not applicable |
| BCDInput.2.Tens | BCD2 Tens (MSD) | uint8 | 2edf | 11999 | Not applicable |
| BCDInput.2.Units | BCD2 Units (LSD) | uint8 | 2ede | 11998 | Not applicable |
| Channel.1.Alarm1.Acknowledge | 1 = Acknowledge alarm | bool | 01b0 | 432 | Not applicable |
| Channel.1.Alarm1.Acknowledgement | 1 = Alarm acknowledged | bool | 1850 | 6224 | Not applicable |
| Channel.1.Alarm1.Active | 1 = Alarm source active, or safe but not ack'd | bool | 184b | 6219 | Not applicable |
| Channel.1.Alarm1.Amount | Alarm amount | float32 | 1848 | 6216 | Same as Channel.1.Main.PV |
| Channel.1.Alarm1.AverageTime | Average time | time_t | 184a | 6218 | Set by Network.Modbus.TimeFormat |
| Channel.1.Alarm1.Block | Blocking enable (0 = Off; 1 = On) | uint8 | 1842 | 6210 | Not applicable |
| Channel.1.Alarm1.ChangeTime | Change time (0 = Per second; 1= Per minute; 2 =Per hour) | uint8 | 1849 | 6217 | Not applicable |
| Channel.1.Alarm1.Deviation | Alarm deviation | float32 | 1847 | 6215 | Same as Channel.1.Main.PV |
| Channel.1.Alarm1.Dwell | Alarm dwell | time_t | 1845 | 6213 | Set by Network.Modbus.TimeFormat |
| Channel.1.Alarm1.Hysteresis | Alarm hysteresis | float32 | 1844 | 6212 | Same as Channel.1.Main.PV |
| Channel.1.Alarm1.Inactive | 1 = the alarm is safe and acknowledged | bool | 184e | 6222 | Not applicable |
| Channel.1.Alarm1.Inhibit | 1 = the alarm is inhibited | bool | 1851 | 6225 | Not applicable |
| Channel.1.Alarm1.Latch | Alarm latch type<br>0 = None 1 = Auto<br>2 = Manual 3 = Trigger | uint8 | 1841 | 6209 | Not applicable |
| Channel.1.Alarm1.NotAcknowledged | 1 = the alarm has not been acknowledged | bool | 184f | 6223 | Not applicable |
| Channel.1.Alarm1.Reference | Alarm reference | float32 | 1846 | 6214 | Same as Channel.1.Main.PV |
| Channel.1.Alarm1.Status | Alarm status<br>0 = Off 1 = Active<br>2 = Safe not acknowledged 3 = Active not acknowledged | uint8 | 0102 | 258 | Not applicable |
| Channel.1.Alarm1.Threshold | Alarm threshold | float32 | 1843 | 6211 | Same as Channel.1.Main.PV |
| Channel.1.Alarm1.Type | Alarn type<br>0 = None 1 = Abs High 2 = Abs Low<br>3 = Dev high 4 = Dev Low 5 = Dev band<br>6 = ROC rising 7 = ROC falling 10 = Dig Off<br>11 = Dig high 12 = Dig Low | uint8 | 1840 | 6208 | Not applicable |
| Channel.1.Alarm2.Acknowledge | 1 = Acknowledge alarm | bool | 01b1 | 433 | Not applicable |
| Channel.1.Alarm2.Acknowledgement | 1 = Alarm acknowledged | bool | 1870 | 6256 | Not applicable |
| Channel.1.Alarm2.Active | 1 = Alarm source active, or safe but not ack'd | bool | 186b | 6251 | Not applicable |
| Channel.1.Alarm2.Amount | Alarm amount | float32 | 1868 | 6248 | Same as Channel.1.Main.PV |
| Channel.1.Alarm2.AverageTime | Average time | time_t | 186a | 6250 | Set by Network.Modbus.TimeFormat |
| Channel.1.Alarm2.Block | Blocking enable (0 = Off; 1 = On) | uint8 | 1862 | 6242 | Not applicable |
| Channel.1.Alarm2.ChangeTime | Change time (0 = Per second; 1= Per minute; 2 =Per hour) | uint8 | 1869 | 6249 | Not applicable |
| Channel.1.Alarm2.Deviation | Alarm deviation | float32 | 1867 | 6247 | Same as Channel.1.Main.PV |
| Channel.1.Alarm2.Dwell | Alarm dwell | time_t | 1865 | 6245 | Set by Network.Modbus.TimeFormat |
| Channel.1.Alarm2.Hysteresis | Alarm hysteresis | float32 | 1864 | 6244 | Same as Channel.1.Main.PV |
| Channel.1.Alarm2.Inactive | 1 = the alarm is safe and acknowledged | bool | 186e | 6254 | Not applicable |
| Channel.1.Alarm2.Inhibit | 1 = the alarm is inhibited | bool | 1871 | 6257 | Not applicable |
| Channel.1.Alarm2.Latch | Configures the latching type of the alarm (As Alarm1.Latch) | uint8 | 1861 | 6241 | Not applicable |
| Channel.1.Alarm2.NotAcknowledged | 1 = the alarm has not been acknowledged | bool | 186f | 6255 | Not applicable |
| Channel.1.Alarm2.Reference | Alarm reference | float32 | 1866 | 6246 | Same as Channel.1.Main.PV |
| Channel.1.Alarm2.Status | As Alarm1.Status | uint8 | 0103 | 259 | Not applicable |
| Channel.1.Alarm2.Threshold | Alarm threshold | float32 | 1863 | 6243 | Same as Channel.1.Main.PV |
| Channel.1.Alarm2.Type | Alarm type (as Alarm1.Type | uint8 | 1860 | 6240 | Not applicable |
| Channel.1.Main.CJType | Cold junction compensation type<br>0 = None 1 = Internal 2 = External<br>3 = Remote (Ch1) 4 = Remote (Ch2) 5 = Remote (Ch3)<br>6 = Remote (Ch4) | uint8 | 180c | 6156 | Not applicable |
| Channel.1.Main.CloseString | Close String | string_t | 4990 | 18832 | Not applicable |
| Channel.1.Main.Descriptor | Text string to describe the channel | string_t | 4900 | 18688 | Not applicable |
| Channel.1.Main.ExtCJTemp | External CJ temperature | float32 | 180d | 6157 | 1dp |
| Channel.1.Main.FaultResponse | Fault response. 0 = none; 1 = Drive high; 2 = Drive low | uint8 | 1810 | 6160 | Not applicable |
| Channel.1.Main.Filter | Filter time constant | float32 | 180e | 6158 | 1dp |
| Channel.1.Main.InputHigh | Input range high value | float32 | 1804 | 6148 | 1dp |
| Channel.1.Main.InputLow | Input range low value | float32 | 1803 | 6147 | 1dp |
| Channel.1.Main.InternalCJTemp | Channel internal cold junction temperature | float32 | 1815 | 6165 | 1dp |
| Channel.1.Main.IPAdjustState | Input Adjust state (0 = Unadjusted; 1 = Adjusted) | bool | 1816 | 6166 | Not applicable |
| Channel.1.Main.IPAdjustState2 | Secondary Input Adjust state (0 = Unadjusted; 1 = Adjusted) | bool | 181c | 6172 | Not applicable |
| Channel.1.Main.LinType | Linearisation type<br>0 =Type B 1 = Type C 2 = Type D<br>3 = Type E 4 = Type G2 5 = Type J<br>6 =Type K 7 = Type L 8 = Type N<br>9 = Type R 10 = Type S 11 = Type T<br>12 = Type U 13 = NiMoNiCo 14 = Platinel<br>15 = NiNiMo 16 = Pt20RhPt40Rh 17 = User 1<br>18 = User 2 19 = User 3 20 = User 4<br>21 = Cu10 22 = Pt100 23 = Pt100A<br>24 = JPT100 25 = Ni100 26 = Ni120<br>27 = Cu53 28 = Linear 29 = Sqrt<br>30 = $x^{3/2}$ 32 = $x^{5/2}$ | uint8 | 1806 | 6150 | Not applicable |
| Channel.1.Main.MeasuredValue | Input value before linearisation, scaling, adjust etc. | float32 | 1814 | 6164 | Set by Channel.1.Main.Resolution |
| Channel.1.Main.MeasuredValue2 | Measured value of the secondary input | float32 | 1819 | 6169 | Set by Channel.1.Main.Resolution |
| Channel.1.Main.Offset | Fixed value to be added to/subtracted from PV | float32 | 1817 | 6167 | 3dp |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| Channel.1.Main.Offset2 | Secondary input offset (as above). | float32 | 1818 | 6168 | 3dp |
| Channel.1.Main.OpenString | Open String | string_t | 496c | 18796 | Not applicable |
| Channel.1.Main.PV | The process variable (output) of the channel | float32 | 0100 | 256 | Set by Channel.1.Main.Resolution |
| Channel.1.Main.PV2 | The secondary input process variable (output) of the channel | float32 | 0110 | 272 | Set by Channel.1.Main.Resolution |
| Channel.1.Main.RangeHigh | Range high value | float32 | 1808 | 6152 | Set by Channel.1.Main.Resolution |
| Channel.1.Main.RangeLow | Range low value | float32 | 1807 | 6151 | Set by Channel.1.Main.Resolution |
| Channel.1.Main.RangeUnits | Range units: 0 = °C; 1 = °F; 2 = Kelvins | uint8 | 1809 | 6153 | Not applicable |
| Channel.1.Main.Resolution | Specifies the resolution/number of decimal places | uint8 | 1801 | 6145 | Not applicable |
| Channel.1.Main.ScaleHigh | Scale high value | float32 | 180b | 6155 | Set by Channel.1.Main.Resolution |
| Channel.1.Main.ScaleHigh2 | Scale high value for the secondary input | float32 | 181b | 6171 | Set by Channel.1.Main.Resolution |
| Channel.1.Main.ScaleLow | Scale low value | float32 | 180a | 6154 | Set by Channel.1.Main.Resolution |
| Channel.1.Main.ScaleLow2 | Scale low value for the secondary input | float32 | 181a | 6170 | Set by Channel.1.Main.Resolution |
| Channel.1.Main.SensorBreakType | Sensor break type: 0 =Off; 1 = Low; 2 = High | uint8 | 180f | 6159 | Not applicable |
| Channel.1.Main.SensorBreakVal | Sensor break value | uint8 | 1811 | 6161 | Not applicable |
| Channel.1.Main.Shunt | Shunt value (Ohms) | float32 | 1805 | 6149 | 2dp |
| Channel.1.Main.Status | The PV (output) status<br>0 =Good 1 = Off 2 = Over range<br>3 = Under range 4 = HW error 5 = Ranging<br>6 = Overflow 7 = bad 8 = HW exceeded<br>9 = No data 12 = Comm channel error | uint8 | 0101 | 257 | Not applicable |
| Channel.1.Main.Status2 | The secondary input PV (output) status (as above) | uint8 | 0111 | 273 | Not applicable |
| Channel.1.Main.TestSignal | Channel test waveform<br>0 = Triangle 5hr 1 = Triangle 40 min<br>2 = Triangle 4 min 3 = Triangle 40 sec<br>4 = Sine 5 hr 5 = Sine 40 min<br>6 = Sine 4 min 7 = Sine 40 sec | uint8 | 1802 | 6146 | Not applicable |
| Channel.1.Main.Type | Specifies the type of channel<br>0 = Off 1 = TC 2 = mV<br>3 = V 4 = mA 5 = RTD<br>6 = Digital 7 = Test 8 = Ohms<br>9 = Dual mV 10 = Dual mA 11 = Dual TC | uint8 | 1800 | 6144 | Not applicable |
| Channel.1.Main.Units | Units descriptor | string_t | 4915 | 18709 | Not applicable |
| Channel.1.Trend.Colour | Configures the trend colour for this channeluint8<br>0 = Red 1 = Blue 2 = Green<br>3 = Honey 4 = Violet 5 = Russet<br>6 = Dark blue 7 = Jade 8 = Magenta<br>9 = Dusky rose 10 = Yellow 11 = Powder blue<br>12 = Dark red 13 = Avocado 14 = Indigo<br>15 = Dark brown 16 = Aegean 17 = Cyan<br>18 = Aubergine 19 = Dark orange 20 = Pale yellow<br>21 = Hyacinth 22 = Dark green 23 = Sugar pink<br>24 = Bluebell 25 = Orange 26 = Pink<br>27 = Buttersilk 28 = Terracotta 29 = Blue babe<br>30 = Lime 31 = Blue jive 32 = Cucumber<br>33 = Eurogreen 34 = Wheatgerm 35 = Sea Blue<br>36 = Ginger 37 = Aqua pool 38 = Pale red<br>39 = Pale blue 40 = Lilac 41 = Sky blue<br>42 = Wild moss 43 = Turquoise 44 = Pale green<br>45 = Coffee 49 = Dark Grey 53 = Light grey | 1820 | | 6176 | Not applicable |
| Channel.1.Trend.SpanHigh | Specifies the highest PV (output value) to be displayed | float32 | 1822 | 6178 | Same as Channel.1.Main.PV |
| Channel.1.Trend.SpanLow | Specifies the lowest PV (output value) to be displayed | float32 | 1821 | 6177 | Same as Channel.1.Main.PV |
| | | | | | |
| Channel.2.Alarm1.Acknowledge | 1 = acknowledge alarm | bool | 01b2 | 434 | Not applicable |
| Channel.2.Alarm1.Acknowledgement | 1 = alarm acknowledged | bool | 18d0 | 6352 | Not applicable |
| Channel.2.Alarm1.Active | 1 = alarm source active, or safe but not ack'd | bool | 18cb | 6347 | Not applicable |
| Channel.2.Alarm1.Amount | Rate-of-change alarm 'Amount' | float32 | 18c8 | 6344 | Same as Channel.2.Main.PV |
| Channel.2.Alarm1.AverageTime | Rate-of-change alarm 'Average time' | time_t | 18ca | 6346 | Set by Network.Modbus.TimeFormat |
| Channel.2.Alarm1.Block | 0 = Blocking alarms off; 1 = Blocking alarms on | uint8 | 18c2 | 6338 | Not applicable |
| Channel.2.Alarm1.ChangeTime | Rate-of-change alarm 'Change Time' | uint8 | 18c9 | 6345 | Not applicable |
| Channel.2.Alarm1.Deviation | Deviation alarm 'Deviation Value' | float32 | 18c7 | 6343 | Same as Channel.2.Main.PV |
| Channel.2.Alarm1.Dwell | Alarm dwell time | time_t | 18c5 | 6341 | Set by Network.Modbus.TimeFormat |
| Channel.2.Alarm1.Hysteresis | Alarm hysteresis value | float32 | 18c4 | 6340 | Same as Channel.2.Main.PV |
| Channel.2.Alarm1.Inactive | 1 = alarm source safe and ack'd (if necessary) | bool | 18ce | 6350 | Not applicable |
| Channel.2.Alarm1.Inhibit | 1 = Alarm inhibited | bool | 18d1 | 6353 | Not applicable |
| Channel.2.Alarm1.Latch | Alarm latch type (as for Channel.1.Alarm1) | uint8 | 18c1 | 6337 | Not applicable |
| Channel.2.Alarm1.NotAcknowledged | 1 = alarm has not been acknowledged | bool | 18cf | 6351 | Not applicable |
| Channel.2.Alarm1.Reference | Deviation alarm 'Reference' value | float32 | 18c6 | 6342 | Same as Channel.2.Main.PV |
| Channel.2.Alarm1.Status | Alarm status (as for Channel.1.Alarm1) | uint8 | 0106 | 262 | Not applicable |
| Channel.2.Alarm1.Threshold | Alarm trigger threshold | float32 | 18c3 | 6339 | Same as Channel.2.Main.PV |
| Channel.2.Alarm1.Type | Alarn type (as for Channel.1.Alarm1) | uint8 | 18c0 | 6336 | Not applicable |
| Channel.2.Alarm2.Acknowledge | 1 = acknowledge alarm | bool | 01b3 | 435 | Not applicable |
| Channel.2.Alarm2.Acknowledgement | 1 = alarm acknowledged | bool | 18f0 | 6384 | Not applicable |
| Channel.2.Alarm2.Active | 1 = alarm source active, or safe but not ack'd | bool | 18eb | 6379 | Not applicable |
| Channel.2.Alarm2.Amount | Rate-of-change alarm 'Amount' | float32 | 18e8 | 6376 | Same as Channel.2.Main.PV |
| Channel.2.Alarm2.AverageTime | Rate-of-change alarm 'Average time' | time_t | 18ea | 6378 | Set by Network.Modbus.TimeFormat |
| Channel.2.Alarm2.Block | 0 = Blocking alarms off; 1 = Blocking alarms on | uint8 | 18e2 | 6370 | Not applicable |
| Channel.2.Alarm2.ChangeTime | Rate-of-change alarm 'Change Time' | uint8 | 18e9 | 6377 | Not applicable |
| Channel.2.Alarm2.Deviation | Deviation alarm 'Deviation Value' | float32 | 18e7 | 6375 | Same as Channel.2.Main.PV |
| Channel.2.Alarm2.Dwell | Alarm dwell time | time_t | 18e5 | 6373 | Set by Network.Modbus.TimeFormat |
| Channel.2.Alarm2.Hysteresis | Alarm hysteresis value | float32 | 18e4 | 6372 | Same as Channel.2.Main.PV |
| Channel.2.Alarm2.Inactive | 1 = alarm source safe and ack'd (if necessary) | bool | 18ee | 6382 | Not applicable |
| Channel.2.Alarm2.Inhibit | 1 = Alarm inhibited | bool | 18f1 | 6385 | Not applicable |
| Channel.2.Alarm2.Latch | Alarm latch type (as for Channel.1.Alarm1) | uint8 | 18e1 | 6369 | Not applicable |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| Channel.2.Alarm2.NotAcknowledged | 1 = alarm has not been acknowledged | bool | 18ef | 6383 | Not applicable |
| Channel.2.Alarm2.Reference | Deviation alarm 'Reference' value | float32 | 18e6 | 6374 | Same as Channel.2.Main.PV |
| Channel.2.Alarm2.Status | Alarm status (as for Channel.1.Alarm1) | uint8 | 0107 | 263 | Not applicable |
| Channel.2.Alarm2.Threshold | Alarm trigger threshold | float32 | 18e3 | 6371 | Same as Channel.2.Main.PV |
| Channel.2.Alarm2.Type | Alarn type (as for Channel.1.Alarm1) | uint8 | 18e0 | 6368 | Not applicable |
| Channel.2.Main.CJType | Cold junction compensation type (as for Channel.1.Main) | uint8 | 188c | 6284 | Not applicable |
| Channel.2.Main.CloseString | Close String | string_t | 4999 | 18841 | Not applicable |
| Channel.2.Main.Descriptor | Text string to describe the channel | string_t | 491b | 18715 | Not applicable |
| Channel.2.Main.ExtCJTemp | External CJ temperature | float32 | 188d | 6285 | 1dp |
| Channel.2.Main.FaultResponse | Input fault response | uint8 | 1890 | 6288 | Not applicable |
| Channel.2.Main.Filter | Filter time constant | float32 | 188e | 6286 | 1dp |
| Channel.2.Main.InputHigh | Input range high value | float32 | 1884 | 6276 | 1dp |
| Channel.2.Main.InputLow | Input range low value | float32 | 1883 | 6275 | 1dp |
| Channel.2.Main.InternalCJTemp | Channel 2 internal cold junction temperature | float32 | 1895 | 6293 | 1dp |
| Channel.2.Main.IPAdjustState | Input Adjust state (0 = Unadjusted; 1 =Adjusted) | bool | 1896 | 6294 | Not applicable |
| Channel.2.Main.IPAdjustState2 | Secondary Input Adjust state (0 = Unadjusted; 1 = Adjusted) | bool | 189c | 6300 | Not applicable |
| Channel.2.Main.LinType | Linearisation type (as for Channel.1.Main) | uint8 | 1886 | 6278 | Not applicable |
| Channel.2.Main.MeasuredValue | Input value before linearisation, scaling, adjust etc. | float32 | 1894 | 6292 | Set by Channel.2.Main.Resolution |
| Channel.2.Main.MeasuredValue2 | Measured value of the secondary input | float32 | 1899 | 6297 | Set by Channel.2.Main.Resolution |
| Channel.2.Main.Offset | Fixed value to be added to/subtracted from PV | float32 | 1897 | 6295 | 3dp |
| Channel.2.Main.Offset2 | Secondary input offset | float32 | 1898 | 6296 | 3dp |
| Channel.2.Main.OpenString | Open String | string_t | 4975 | 18805 | Not applicable |
| Channel.2.Main.PV | The output (displayed) value of the channel. | float32 | 0104 | 260 | Set by Channel.2.Main.Resolution |
| Channel.2.Main.PV2 | The secondary input process variable (output) of the channel | float32 | 0114 | 276 | Set by Channel.2.Main.Resolution |
| Channel.2.Main.RangeHigh | Range high value | float32 | 1888 | 6280 | Set by Channel.2.Main.Resolution |
| Channel.2.Main.RangeLow | Range low value | float32 | 1887 | 6279 | Set by Channel.2.Main.Resolution |
| Channel.2.Main.RangeUnits | Range units (as channel.1.Main) | uint8 | 1889 | 6281 | Not applicable |
| Channel.2.Main.Resolution | Specifies the resolution/number of decimal places | uint8 | 1881 | 6273 | Not applicable |
| Channel.2.Main.ScaleHigh | Scale high value | float32 | 188b | 6283 | Set by Channel.2.Main.Resolution |
| Channel.2.Main.ScaleHigh2 | Scale high value for the secondary input | float32 | 189b | 6299 | Set by Channel.2.Main.Resolution |
| Channel.2.Main.ScaleLow | Scale low value | float32 | 188a | 6282 | Set by Channel.2.Main.Resolution |
| Channel.2.Main.ScaleLow2 | Scale low value for the secondary input | float32 | 189a | 6298 | Set by Channel.2.Main.Resolution |
| Channel.2.Main.SensorBreakType | Sensor break type (as for Channel.1.Main) | uint8 | 188f | 6287 | Not applicable |
| Channel.2.Main.SensorBreakVal | Sensor break value | uint8 | 1891 | 6289 | Not applicable |
| Channel.2.Main.Shunt | Shunt value in Ohms | float32 | 1885 | 6277 | 2dp |
| Channel.2.Main.Status | Channel status (as for Channel.1.Main.Status) | uint8 | 0105 | 261 | Not applicable |
| Channel.2.Main.Status2 | The secondary input PV (output) status (as above) | uint8 | 0115 | 277 | Not applicable |
| Channel.2.Main.TestSignal | Channel test waveform (as for Channel.1.Main) | uint8 | 1882 | 6274 | Not applicable |
| Channel.2.Main.Type | Channel function (as for Channel.1.Main.Type) | uint8 | 1880 | 6272 | Not applicable |
| Channel.2.Main.Units | Channel units string | string_t | 4930 | 18736 | Not applicable |
| Channel.2.Trend.Colour | Trend colour (as for Channel.1.Trend.Colour) | uint8 | 18a0 | 6304 | Not applicable |
| Channel.2.Trend.SpanHigh | Specifies the highest PV (output value) to be displayed | float32 | 18a2 | 6306 | Same as Channel.2.Main.PV |
| Channel.2.Trend.SpanLow | Specifies the lowest PV (output value) to be displayed | float32 | 18a1 | 6305 | Same as Channel.2.Main.PV |
| Channel.3.Alarm1.Acknowledgement | 1 = alarm acknowledged | bool | 1950 | 6480 | Not applicable |
| Channel.3.Alarm1.Active | 1 = alarm source active, or safe but not ack'd | bool | 194b | 6475 | Not applicable |
| Channel.3.Alarm1.Amount | Rate-of-change alarm 'Amount' | float32 | 1948 | 6472 | Same as Channel.3.Main.PV |
| Channel.3.Alarm1.AverageTime | Rate-of-change alarm 'Average time' | time_t | 194a | 6474 | Set by Network.Modbus.TimeFormat |
| Channel.3.Alarm1.Block | 0 = Blocking alarms off; 1 = Blocking alarms on. | uint8 | 1942 | 6466 | Not applicable |
| Channel.3.Alarm1.ChangeTime | Rate-of-change alarm 'Change Time' | uint8 | 1949 | 6473 | Not applicable |
| Channel.3.Alarm1.Deviation | Deviation alarm 'Deviation Value' | float32 | 1947 | 6471 | Same as Channel.3.Main.PV |
| Channel.3.Alarm1.Dwell | Alarm dwell time | time_t | 1945 | 6469 | Set by Network.Modbus.TimeFormat |
| Channel.3.Alarm1.Hysteresis | Alarm hysteresis value | float32 | 1944 | 6468 | Same as Channel.3.Main.PV |
| Channel.3.Alarm1.Inactive | 1 = alarm source safe and ack'd (if necessary) | bool | 194e | 6478 | Not applicable |
| Channel.3.Alarm1.Inhibit | 1 = alarm inhibited | bool | 1951 | 6481 | Not applicable |
| Channel.3.Alarm1.Latch | Alarm latch type (as for Channel.1.Alarm1) | uint8 | 1941 | 6465 | Not applicable |
| Channel.3.Alarm1.NotAcknowledged | 1 = alarm has not been acknowledged | bool | 194f | 6479 | Not applicable |
| Channel.3.Alarm1.Reference | Deviation alarm 'Reference' value | float32 | 1946 | 6470 | Same as Channel.3.Main.PV |
| Channel.3.Alarm1.Status | Alarm status (as for Channel.1.Alarm1) | uint8 | 010a | 266 | Not applicable |
| Channel.3.Alarm1.Threshold | Alarm trigger threshold | float32 | 1943 | 6467 | Same as Channel.3.Main.PV |
| Channel.3.Alarm1.Type | Alarn type (as for Channel.1.Alarm1) | uint8 | 1940 | 6464 | Not applicable |
| Channel.3.Alarm2.Acknowledge | 1 = acknowledge alarm | bool | 01b5 | 437 | Not applicable |
| Channel.3.Alarm2.Acknowledgement | 1 = alarm acknowledged | bool | 1970 | 6512 | Not applicable |
| Channel.3.Alarm2.Active | 1 = alarm source active, or safe but not ack'd | bool | 196b | 6507 | Not applicable |
| Channel.3.Alarm2.Amount | Rate-of-change alarm 'Amount' | float32 | 1968 | 6504 | Same as Channel.3.Main.PV |
| Channel.3.Alarm2.AverageTime | Rate-of-change alarm 'Average time' | time_t | 196a | 6506 | Set by Network.Modbus.TimeFormat |
| Channel.3.Alarm2.Block | 0 = Blocking alarms off; 1 = Blocking alarms on. | uint8 | 1962 | 6498 | Not applicable |
| Channel.3.Alarm2.ChangeTime | Rate-of-change alarm 'Change Time' | uint8 | 1969 | 6505 | Not applicable |
| Channel.3.Alarm2.Deviation | Deviation alarm 'Deviation Value' | float32 | 1967 | 6503 | Same as Channel.3.Main.PV |
| Channel.3.Alarm2.Dwell | Alarm dwell time | time_t | 1965 | 6501 | Set by Network.Modbus.TimeFormat |
| Channel.3.Alarm2.Hysteresis | Alarm hysteresis value | float32 | 1964 | 6500 | Same as Channel.3.Main.PV |
| Channel.3.Alarm2.Inactive | 1 = alarm source safe and ack'd (if necessary) | bool | 196e | 6510 | Not applicable |
| Channel.3.Alarm2.Inhibit | 1 = Alarm inhibited | bool | 1971 | 6513 | Not applicable |
| Channel.3.Alarm2.Latch | Alarm latch type (as for Channel.1.Alarm1) | uint8 | 1961 | 6497 | Not applicable |
| Channel.3.Alarm2.NotAcknowledged | 1 = alarm has not been acknowledged | bool | 196f | 6511 | Not applicable |
| Channel.3.Alarm2.Reference | Deviation alarm 'Reference' value | float32 | 1966 | 6502 | Same as Channel.3.Main.PV |
| Channel.3.Alarm2.Status | Alarm status (as for Channel.1.Alarm1) | uint8 | 010b | 267 | Not applicable |
| Channel.3.Alarm2.Threshold | Alarm trigger threshold | float32 | 1963 | 6499 | Same as Channel.3.Main.PV |
| Channel.3.Alarm2.Type | Alarn type (as for Channel.1.Alarm1) | uint8 | 1960 | 6496 | Not applicable |
| Channel.3.Main.CJType | Cold junction compensation type (as for Channel.1.Main) | uint8 | 190c | 6412 | Not applicable |
| Channel.3.Main.CloseString | Close String | string_t | 49a2 | 18850 | Not applicable |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| Channel.3.Main.Descriptor | Text string to describe the channel | string_t | 4936 | 18742 | Not applicable |
| Channel.3.Main.ExtCJTemp | External CJ temperature | float32 | 190d | 6413 | 1dp |
| Channel.3.Main.FaultResponse | Input fault response (As for Channel.1.Main) | uint8 | 1910 | 6416 | Not applicable |
| Channel.3.Main.Filter | Filter time constant | float32 | 190e | 6414 | 1dp |
| Channel.3.Main.InputHigh | Input range maximum value | float32 | 1904 | 6404 | 1dp |
| Channel.3.Main.InputLow | Input range minimum value | float32 | 1903 | 6403 | 1dp |
| Channel.3.Main.InternalCJTemp | Channel internal cold junction temperature | float32 | 1915 | 6421 | 1dp |
| Channel.3.Main.IPAdjustState | Input Adjust state (0 = Unadjusted; 1 = Adjusted) | bool | 1916 | 6422 | Not applicable |
| Channel.3.Main.IPAdjustState2 | Secondary Input Adjust state (0 = Unadjusted; 1 = Adjusted) | bool | 191c | 6428 | Not applicable |
| Channel.3.Main.LinType | Linearisation type (as for Channel.1.Main.LinType) | uint8 | 1906 | 6406 | Not applicable |
| Channel.3.Main.MeasuredValue | Input value before linearisation, scaling, adjust etc. | float32 | 1914 | 6420 | Set by Channel.3.Main.Resolution |
| Channel.3.Main.MeasuredValue2 | Measured value of the secondary input | float32 | 1919 | 6425 | Set by Channel.3.Main.Resolution |
| Channel.3.Main.Offset | Input offset | float32 | 1917 | 6423 | 3dp |
| Channel.3.Main.Offset2 | Secondary input offset | float32 | 1918 | 6424 | 3dp |
| Channel.3.Main.OpenString | Open String | string_t | 497e | 18814 | Not applicable |
| Channel.3.Main.PV | The output (displayed) value of the channel. | float32 | 0108 | 264 | Set by Channel.3.Main.Resolution |
| Channel.3.Main.PV2 | The secondary input process variable (output) of the channel | float32 | 0118 | 280 | Set by Channel.3.Main.Resolution |
| Channel.3.Main.RangeHigh | Range high value | float32 | 1908 | 6408 | Set by Channel.3.Main.Resolution |
| Channel.3.Main.RangeLow | Range low value | float32 | 1907 | 6407 | Set by Channel.3.Main.Resolution |
| Channel.3.Main.RangeUnits | Range units | uint8 | 1909 | 6409 | Not applicable |
| Channel.3.Main.Resolution | Specifies the resolution/number of decimal places | uint8 | 1901 | 6401 | Not applicable |
| Channel.3.Main.ScaleHigh | Scale high value | float32 | 190b | 6411 | Set by Channel.3.Main.Resolution |
| Channel.3.Main.ScaleHigh2 | Scale high value for the secondary input | float32 | 191b | 6427 | Set by Channel.3.Main.Resolution |
| Channel.3.Main.ScaleLow | Scale low value | float32 | 190a | 6410 | Set by Channel.3.Main.Resolution |
| Channel.3.Main.ScaleLow2 | Scale low value for the secondary input | float32 | 191a | 6426 | Set by Channel.3.Main.Resolution |
| Channel.3.Main.SensorBreakType | Sensor break type (as for Channel.1.Main) | uint8 | 190f | 6415 | Not applicable |
| Channel.3.Main.SensorBreakVal | Sensor break value | uint8 | 1911 | 6417 | Not applicable |
| Channel.3.Main.Shunt | Shunt value in Ohms | float32 | 1905 | 6405 | 2dp |
| Channel.3.Main.Status | Channel status (as for Channel.1.Main.Status) | uint8 | 0109 | 265 | Not applicable |
| Channel.3.Main.Status2 | The secondary input PV (output) status | uint8 | 0119 | 281 | Not applicable |
| Channel.3.Main.TestSignal | Channel test waveform (as for Channel.1.Main) | uint8 | 1902 | 6402 | Not applicable |
| Channel.3.Main.Type | Channel function (as for Channel.1.Main.Type) | uint8 | 1900 | 6400 | Not applicable |
| Channel.3.Main.Units | Units descriptor | string_t | 494b | 18763 | Not applicable |
| Channel.3.Trend.Colour | Trend colour (as for Channel.1.Trend.Colour) | uint8 | 1920 | 6432 | Not applicable |
| Channel.3.Trend.SpanHigh | Specifies the highest PV (output value) to be displayed | float32 | 1922 | 6434 | Same as Channel.3.Main.PV |
| Channel.3.Trend.SpanLow | Specifies the lowest PV (output value) to be displayed | float32 | 1921 | 6433 | Same as Channel.3.Main.PV |
| Channel.4.Alarm1.Acknowledge | 1 = acknowledge alarm | bool | 01b6 | 438 | Not applicable |
| Channel.4.Alarm1.Acknowledgement | 1 = alarm acknowledged | bool | 19d0 | 6608 | Not applicable |
| Channel.4.Alarm1.Active | 1 = alarm source active, or safe but not ack'd | bool | 19cb | 6603 | Not applicable |
| Channel.4.Alarm1.Amount | Rate-of-change alarm 'Amount' | float32 | 19c8 | 6600 | Same as Channel.4.Main.PV |
| Channel.4.Alarm1.AverageTime | Rate-of-change alarm 'Average time' | time_t | 19ca | 6602 | Set by Network.Modbus.TimeFormat |
| Channel.4.Alarm1.Block | 0 = Blocking alarms off; 1 = Blocking alarms on. | uint8 | 19c2 | 6594 | Not applicable |
| Channel.4.Alarm1.ChangeTime | Rate-of-change alarm 'Change Time' | uint8 | 19c9 | 6601 | Not applicable |
| Channel.4.Alarm1.Deviation | Deviation alarm 'Deviation Value' | float32 | 19c7 | 6599 | Same as Channel.4.Main.PV |
| Channel.4.Alarm1.Dwell | Alarm dwell time | time_t | 19c5 | 6597 | Set by Network.Modbus.TimeFormat |
| Channel.4.Alarm1.Hysteresis | Alarm hysteresis value | float32 | 19c4 | 6596 | Same as Channel.4.Main.PV |
| Channel.4.Alarm1.Inactive | 1 = alarm source safe and ack'd (if necessary) | bool | 19ce | 6606 | Not applicable |
| Channel.4.Alarm1.Inhibit | 1 = alarm inhibited | bool | 19d1 | 6609 | Not applicable |
| Channel.4.Alarm1.Latch | Alarm latch type (as for Channel.1.Alarm1) | uint8 | 19c1 | 6593 | Not applicable |
| Channel.4.Alarm1.NotAcknowledged | 1 = alarm has not been acknowledged | bool | 19cf | 6607 | Not applicable |
| Channel.4.Alarm1.Reference | Deviation alarm 'Reference' value | float32 | 19c6 | 6598 | Same as Channel.4.Main.PV |
| Channel.4.Alarm1.Status | Alarm status (as for Channel.1.Alarm1) | uint8 | 010e | 270 | Not applicable |
| Channel.4.Alarm1.Threshold | Alarm trigger threshold | float32 | 19c3 | 6595 | Same as Channel.4.Main.PV |
| Channel.4.Alarm1.Type | Alarn type (as for Channel.1.Alarm1) | uint8 | 19c0 | 6592 | Not applicable |
| Channel.4.Alarm2.Acknowledge | 1 = acknowledge alarm | bool | 01b7 | 439 | Not applicable |
| Channel.4.Alarm2.Acknowledgement | 1 = alarm acknowledged | bool | 19f0 | 6640 | Not applicable |
| Channel.4.Alarm2.Active | 1 = alarm source active, or safe but not ack'd | bool | 19eb | 6635 | Not applicable |
| Channel.4.Alarm2.Amount | Rate-of-change alarm 'Amount' | float32 | 19e8 | 6632 | Same as Channel.4.Main.PV |
| Channel.4.Alarm2.AverageTime | Rate-of-change alarm 'Average time' | time_t | 19ea | 6634 | Set by Network.Modbus.TimeFormat |
| Channel.4.Alarm2.Block | 0 = Blocking alarms off; 1 = Blocking alarms on | uint8 | 19e2 | 6626 | Not applicable |
| Channel.4.Alarm2.ChangeTime | Rate-of-change alarm 'Change Time' | uint8 | 19e9 | 6633 | Not applicable |
| Channel.4.Alarm2.Deviation | Deviation alarm 'Deviation Value' | float32 | 19e7 | 6631 | Same as Channel.4.Main.PV |
| Channel.4.Alarm2.Dwell | Alarm dwell time | time_t | 19e5 | 6629 | Set by Network.Modbus.TimeFormat |
| Channel.4.Alarm2.Hysteresis | Alarm hysteresis value | float32 | 19e4 | 6628 | Same as Channel.4.Main.PV |
| Channel.4.Alarm2.Inactive | 1 = alarm source safe and ack'd (if necessary) | bool | 19ee | 6638 | Not applicable |
| Channel.4.Alarm2.Latch | Alarm latch type (as for Channel.1.Alarm1) | uint8 | 19e1 | 6625 | Not applicable |
| Channel.4.Alarm2.NotAcknowledged | 1 = alarm has not been acknowledged | bool | 19ef | 6639 | Not applicable |
| Channel.4.Alarm2.Reference | Deviation alarm 'Reference' value | float32 | 19e6 | 6630 | Same as Channel.4.Main.PV |
| Channel.4.Alarm2.Status | Alarm status (as for Channel.1.Alarm1) | uint8 | 010f | 271 | Not applicable |
| Channel.4.Alarm2.Threshold | Alarm trigger threshold | float32 | 19e3 | 6627 | Same as Channel.4.Main.PV |
| Channel.4.Alarm2.Type | Alarn type (as for Channel.1.Alarm1) | uint8 | 19e0 | 6624 | Not applicable |
| Channel.4.Main.CJType | Cold junction compensation type(as for Channel.1.Main) | uint8 | 198c | 6540 | Not applicable |
| Channel.4.Main.CloseString | Close String | string_t | 49ab | 18859 | Not applicable |
| Channel.4.Main.Descriptor | Text string to describe the channel | string_t | 4951 | 18769 | Not applicable |
| Channel.4.Main.ExtCJTemp | External CJ temperature | float32 | 198d | 6541 | 1dp |
| Channel.4.Main.FaultResponse | Input fault response (as for Channel.1.Main) | uint8 | 1990 | 6544 | Not applicable |
| Channel.4.Main.Filter | Filter time constant | float32 | 198e | 6542 | 1dp |
| Channel.4.Main.InputHigh | Input range maximum value | float32 | 1984 | 6532 | 1dp |
| Channel.4.Main.InputLow | Input range minimum value | float32 | 1983 | 6531 | 1dp |
| Channel.4.Main.InternalCJTemp | Channel internal cold junction temperature | float32 | 1995 | 6549 | 1dp |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| Channel.4.Main.IPAdjustState | Input Adjust state (0 = Unadjusted;1 =Adjusted) | bool | 1996 | 6550 | Not applicable |
| Channel.4.Main.IPAdjustState2 | Secondary Input Adjust state (0 = Unadjusted; 1 = Adjusted | bool | 199c | 6556 | Not applicable |
| Channel.4.Main.LinType | Linearisation type (as for Channel.1.Main.LinType) | uint8 | 1986 | 6534 | Not applicable |
| Channel.4.Main.MeasuredValue | Input value before linearisation, scaling, adjust etc. | float32 | 1994 | 6548 | Set by Channel.4.Main.Resolution |
| Channel.4.Main.MeasuredValue2 | Measured value of the secondary input | float32 | 1999 | 6553 | Set by Channel.4.Main.Resolution |
| Channel.4.Main.Offset | Fixed value to be added to/subtracted from PV | float32 | 1997 | 6551 | 3dp |
| Channel.4.Main.Offset2 | Secondary input offset | float32 | 1998 | 6552 | 3dp |
| Channel.4.Main.OpenString | Open String | string_t | 4987 | 18823 | Not applicable |
| Channel.4.Main.PV | The output (displayed) value of the channel. | float32 | 010c | 268 | Set by Channel.4.Main.Resolution |
| Channel.4.Main.PV2 | The secondary input process variable (output) of the channel | float32 | 011c | 284 | Set by Channel.4.Main.Resolution |
| Channel.4.Main.RangeHigh | Range high value | float32 | 1988 | 6536 | Set by Channel.4.Main.Resolution |
| Channel.4.Main.RangeLow | Range low value | float32 | 1987 | 6535 | Set by Channel.4.Main.Resolution |
| Channel.4.Main.RangeUnits | Range units (as channel.1.Main.RangeUnits) | uint8 | 1989 | 6537 | Not applicable |
| Channel.4.Main.Resolution | Specifies the resolution/number of decimal places | uint8 | 1981 | 6529 | Not applicable |
| Channel.4.Main.ScaleHigh | Scale high value | float32 | 198b | 6539 | Set by Channel.4.Main.Resolution |
| Channel.4.Main.ScaleHigh2 | Scale high value for the secondary input | float32 | 199b | 6555 | Set by Channel.4.Main.Resolution |
| Channel.4.Main.ScaleLow | Scale low value | float32 | 198a | 6538 | Set by Channel.4.Main.Resolution |
| Channel.4.Main.ScaleLow2 | Scale low value for the secondary input | float32 | 199a | 6554 | Set by Channel.4.Main.Resolution |
| Channel.4.Main.SensorBreakType | Sensor break type (as for Channel.1.Main) | uint8 | 198f | 6543 | Not applicable |
| Channel.4.Main.SensorBreakVal | Sensor break value | uint8 | 1991 | 6545 | Not applicable |
| Channel.4.Main.Shunt | Shunt value in Ohms | float32 | 1985 | 6533 | 2dp |
| Channel.4.Main.Status | Channel status (as for Channel.1.Main.Status) | uint8 | 010d | 269 | Not applicable |
| Channel.4.Main.Status2 | The secondary input PV (output) status | uint8 | 011d | 285 | Not applicable |
| Channel.4.Main.TestSignal | Channel test waveform (as for Channel.1.Main.TestSignal) | uint8 | 1982 | 6530 | Not applicable |
| Channel.4.Main.Type | Channel function (as for Channel.1.Main.Type) | uint8 | 1980 | 6528 | Not applicable |
| Channel.4.Main.Units | Units descriptor | string_t | 4966 | 18790 | Not applicable |
| Channel.4.Trend.Colour | Trend colour (as for Channel.1.Trend.Colour)uint8 | 19a0 | | 6560 | Not applicable |
| Channel.4.Trend.SpanHigh | Specifies the highest PV (output value) to be displayed | float32 | 19a2 | 6562 | Same as Channel.4.Main.PV |
| Channel.4.Trend.SpanLow | Specifies the lowest PV (output value) to be displayed | float32 | 19a1 | 6561 | Same as Channel.4.Main.PV |
| | | | | | |
| CustomMessage.Message1 | Custom message No 1 | string_t | 5e00 | 24064 | Not applicable |
| CustomMessage.Message2 | Custom message No 2 | string_t | 5e65 | 24165 | Not applicable |
| CustomMessage.Message3 | Custom message No 3 | string_t | 5eca | 24266 | Not applicable |
| CustomMessage.Message4 | Custom message No 4 | string_t | 5f2f | 24367 | Not applicable |
| CustomMessage.Message5 | Custom message No 5 | string_t | 5f94 | 24468 | Not applicable |
| CustomMessage.Message6 | Custom message No 6 | string_t | 5ff9 | 24569 | Not applicable |
| CustomMessage.Message7 | Custom message No 7 | string_t | 605e | 24670 | Not applicable |
| CustomMessage.Message8 | Custom message No 8 | string_t | 60c3 | 24771 | Not applicable |
| CustomMessage.Message9 | Custom message No 9 | string_t | 6128 | 24872 | Not applicable |
| CustomMessage.Message10 | Custom message No 10 | string_t | 618d | 24973 | Not applicable |
| CustomMessage.Trigger1 | Trigger for custom message No 1 | bool | 28f0 | 10480 | Not applicable |
| CustomMessage.Trigger2 | Trigger for custom message No 2 | bool | 28f1 | 10481 | Not applicable |
| CustomMessage.Trigger3 | Trigger for custom message No 3 | bool | 28f2 | 10482 | Not applicable |
| CustomMessage.Trigger4 | Trigger for custom message No 4 | bool | 28f3 | 10483 | Not applicable |
| CustomMessage.Trigger5 | Trigger for custom message No 5 | bool | 28f4 | 10484 | Not applicable |
| CustomMessage.Trigger6 | Trigger for custom message No 6 | bool | 28f5 | 10485 | Not applicable |
| CustomMessage.Trigger7 | Trigger for custom message No 7 | bool | 28f6 | 10486 | Not applicable |
| CustomMessage.Trigger8 | Trigger for custom message No 8 | bool | 28f7 | 10487 | Not applicable |
| CustomMessage.Trigger9 | Trigger for custom message No 9 | bool | 28f8 | 10488 | Not applicable |
| CustomMessage.Trigger10 | Trigger for custom message No 10 | bool | 28f9 | 10489 | Not applicable |
| | | | | | |
| DCOutput.1A1B_DCOP.FallbackPV | Fallback PV value | float32 | 15c9 | 5577 | Set by DCOutput.1A1B_DCOP.Resolution |
| DCOutput.1A1B_DCOP.MeasuredValue | Measured Value | float32 | 15ca | 5578 | 2dp |
| DCOutput.1A1B_DCOP.OPAdjustState | 0 = Unadjusted, 1 = Adjusted | bool | 15c3 | 5571 | Not applicable |
| DCOutput.1A1B_DCOP.OutputHigh | DC Output High value | float32 | 15c6 | 5574 | 2dp |
| DCOutput.1A1B_DCOP.OutputLow | DC Output Low value | float32 | 15c5 | 5573 | 2dp |
| DCOutput.1A1B_DCOP.PV | DC Output PV | float32 | 15c1 | 5569 | Set by DCOutput.1A1B_DCOP.Resolution |
| DCOutput.1A1B_DCOP.Resolution | Specifies the resolution/number of decimal places | uint8 | 15c4 | 5572 | Not applicable |
| DCOutput.1A1B_DCOP.ScaleHigh | Scale High value | float32 | 15c8 | 5576 | Set by DCOutput.1A1B_DCOP.Resolution |
| DCOutput.1A1B_DCOP.ScaleLow | Scale Low value | float32 | 15c7 | 5575 | Set by DCOutput.1A1B_DCOP.Resolution |
| DCOutput.1A1B_DCOP.Status | PV Status<br>0 = Good 1 = Off 2 = Over range<br>3 = Under range 4 = HW error 5 = Ranging<br>6 = Overflow 7 = Bad 8 = HW exceeded<br>9 = No data 10 = Comms channel error | uint8 | 15c2 | 5570 | Not applicable |
| DCOutput.1A1B_DCOP.Type | DC Output Type (0 = Volts; 1 = mA) | uint8 | 15c0 | 5568 | Not applicable |
| | | | | | |
| DCOutput.2A2B_DCOP.FallbackPV | Fallback PV value | float32 | 15b9 | 5561 | Set by DCOutput.2A2B_DCOP.Resolution |
| DCOutput.2A2B_DCOP.MeasuredValue | Measured Value | float32 | 15ba | 5562 | 2dp |
| DCOutput.2A2B_DCOP.OPAdjustState | 0 = Unadjusted, 1 = Adjusted | bool | 15b3 | 5555 | Not applicable |
| DCOutput.2A2B_DCOP.OutputHigh | DC Output High value | float32 | 15b6 | 5558 | 2dp |
| DCOutput.2A2B_DCOP.OutputLow | DC Output Low value | float32 | 15b5 | 5557 | 2dp |
| DCOutput.2A2B_DCOP.PV | DC Output PV | float32 | 15b1 | 5553 | Set by DCOutput.2A2B_DCOP.Resolution |
| DCOutput.2A2B_DCOP.Resolution | Specifies the resolution/number of decimal places | uint8 | 15b4 | 5556 | Not applicable |
| DCOutput.2A2B_DCOP.ScaleHigh | Scale High value | float32 | 15b8 | 5560 | Set by DCOutput.2A2B_DCOP.Resolution |
| DCOutput.2A2B_DCOP.ScaleLow | Scale Low value | float32 | 15b7 | 5559 | Set by DCOutput.2A2B_DCOP.Resolution |
| DCOutput.2A2B_DCOP.Status | PV Status (as DCOutput.1A1B_DCOP.Status) | uint8 | 15b2 | 5554 | Not applicable |
| DCOutput.2A2B_DCOP.Type | DC Output Type (0 = Volts; 1 = mA) | uint8 | 15b0 | 5552 | Not applicable |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| DCOutput.3A3B_DCOP.FallbackPV | Fallback PV value | float32 | 15a9 | 5545 | Set by DCOutput.3A3B_DCOP.Resolution |
| DCOutput.3A3B_DCOP.MeasuredValue | Measured Value | float32 | 15aa | 5546 | 2dp |
| DCOutput.3A3B_DCOP.OPAdjustState | 0 = Unadjusted, 1 = Adjusted | bool | 15a3 | 5539 | Not applicable |
| DCOutput.3A3B_DCOP.OutputHigh | DC Output High value | float32 | 15a6 | 5542 | 2dp |
| DCOutput.3A3B_DCOP.OutputLow | DC Output Low value | float32 | 15a5 | 5541 | 2dp |
| DCOutput.3A3B_DCOP.PV | DC Output PV | float32 | 15a1 | 5537 | Set by DCOutput.3A3B_DCOP.Resolution |
| DCOutput.3A3B_DCOP.Resolution | Specifies the resolution/number of decimal places | uint8 | 15a4 | 5540 | Not applicable |
| DCOutput.3A3B_DCOP.ScaleHigh | Scale High value | float32 | 15a8 | 5544 | Set by DCOutput.3A3B_DCOP.Resolution |
| DCOutput.3A3B_DCOP.ScaleLow | Scale Low value | float32 | 15a7 | 5543 | Set by DCOutput.3A3B_DCOP.Resolution |
| DCOutput.3A3B_DCOP.Status | PV Status (as DCOutput.1A1B_DCOP.Status) | uint8 | 15a2 | 5538 | Not applicable |
| DCOutput.3A3B_DCOP.Type | DC Output Type (0 = Volts; 1 = mA) | uint8 | 15a0 | 5536 | Not applicable |
| DigitalIO.1A1B.Backlash | Valve positioning backlash compensation (seconds) | float32 | 1508 | 5384 | 1dp |
| DigitalIO.1A1B.Inertia | Inertia value for the valve | float32 | 1507 | 5383 | 1dp |
| DigitalIO.1A1B.Invert | 1 = Invert; 0 = Do not invert | bool | 1503 | 5379 | Not applicable |
| DigitalIO.1A1B.MinOnTime | Time proportioned output minimum on time | float32 | 1502 | 5378 | 2dp |
| DigitalIO.1A1B.ModuleIdent | Module Identification<br>0 = Digital I/O 1 = Relay output 2 = Triac output<br>3 = Digital input 4 = Digital output | uint8 | 150a | 5386 | Not applicable |
| DigitalIO.1A1B.Output | 0 = Output off, 1 = Output on | bool | 1504 | 5380 | Not applicable |
| DigitalIO.1A1B.PV | For contact inputs, 0 = Open, 1 = Closed.<br>For On Off outputs, <0.5 = Drive low, else drive high<br>For Time Proportional outputs, PV = demanded output % | float32 | 1501 | 5377 | 0dp |
| DigitalIO.1A1B.StandbyAction | Valve positioning standby action (0 = Continue; 1 = Freeze). | uint8 | 1509 | 5385 | Not applicable |
| DigitalIO.1A1B.Type | Specifies the type of the digital input / output<br>0 = Contact closure input 1 = On Off output<br>2 = Time proportioniing output 3 = Valve raise<br>4 = Valve lower | uint8 | 1500 | 5376 | Not applicable |
| DigitalIO.2A2B.Backlash | Valve positioning backlash compensation (seconds) | float32 | 1518 | 5400 | 1dp |
| DigitalIO.2A2B.Inertia | Inertia value for the valve | float32 | 1517 | 5399 | 1dp |
| DigitalIO.2A2B.Invert | 1 = Invert; 0 = Do not invert | bool | 1513 | 5395 | Not applicable |
| DigitalIO.2A2B.MinOnTime | Time proportioned output minimum on time | float32 | 1512 | 5394 | 2dp |
| DigitalIO.2A2B.ModuleIdent | As DigitalIO.1A1B.ModuleIdent | uint8 | 151a | 5402 | Not applicable |
| DigitalIO.2A2B.Output | 0 = Output off, 1 = Output on | bool | 1514 | 5396 | Not applicable |
| DigitalIO.2A2B.PV | Digital I/O process value (as DigitalIO.1A1B.PV) | float32 | 1511 | 5393 | 0dp |
| DigitalIO.2A2B.StandbyAction | Valve positioning standby action (0 = Continue; 1 = Freeze). | uint8 | 1519 | 5401 | Not applicable |
| DigitalIO.2A2B.Type | Digital I/O type (as DigitalIO.1A1B.Type). | uint8 | 1510 | 5392 | Not applicable |
| DigitalIO.3A3B.Backlash | Valve positioning backlash compensation (seconds) | float32 | 1538 | 5432 | 1dp |
| DigitalIO.3A3B.Inertia | Inertia value for the valve | float32 | 1537 | 5431 | 1dp |
| DigitalIO.3A3B.Invert | 1 = Invert; 0 = Do not invert | bool | 1533 | 5427 | Not applicable |
| DigitalIO.3A3B.MinOnTime | Time proportioned output minimum on time | float32 | 1532 | 5426 | 2dp |
| DigitalIO.3A3B.ModuleIdent | As DigitalIO.1A1B.ModuleIdent | uint8 | 153a | 5434 | Not applicable |
| DigitalIO.3A3B.Output | 0 = Output off, 1 = Output on | bool | 1534 | 5428 | Not applicable |
| DigitalIO.3A3B.PV | Digital I/O process value (as DigitalIO.1A1B.PV) | float32 | 1531 | 5425 | 0dp |
| DigitalIO.3A3B.StandbyAction | Valve positioning standby action (0 = Continue; 1 = Freeze). | uint8 | 1539 | 5433 | Not applicable |
| DigitalIO.3A3B.Type | Digital I/O type (as DigitalIO.1A1B.Type). | uint8 | 1530 | 5424 | Not applicable |
| DigitalIO.DI_LALC.Backlash | Valve positioning backlash compensation (seconds) | float32 | 1528 | 5416 | 1dp |
| DigitalIO.DI_LALC.Inertia | Inertia value for the valve | float32 | 1527 | 5415 | 1dp |
| DigitalIO.DI_LALC.Invert | 1 = Invert; 0 = Do not invert | bool | 1523 | 5411 | Not applicable |
| DigitalIO.DI_LALC.MinOnTime | Time proportioned output minimum on time | float32 | 1522 | 5410 | 2dp |
| DigitalIO.DI_LALC.ModuleIdent | As DigitalIO.1A1B.ModuleIdent | uint8 | 152a | 5418 | Not applicable |
| DigitalIO.DI_LALC.Output | 0 = Output off, 1 = Output on | bool | 1524 | 5412 | Not applicable |
| DigitalIO.DI_LALC.PV | Digital I/O process value (as DigitalIO.1A1B.PV) | float32 | 1521 | 5409 | 0dp |
| DigitalIO.DI_LALC.StandbyAction | Valve positioning standby action (0 = Continue; 1 = Freeze). | uint8 | 1529 | 5417 | Not applicable |
| DigitalIO.DI_LALC.Type | Digital I/O type (as DigitalIO.1A1B.Type). | uint8 | 1520 | 5408 | Not applicable |
| DigitalIO.DI_LBLC.Backlash | Valve positioning backlash compensation (seconds) | float32 | 1548 | 5448 | 1dp |
| DigitalIO.DI_LBLC.Inertia | Inertia value for the valve | float32 | 1547 | 5447 | 1dp |
| DigitalIO.DI_LBLC.Invert | 1 = Invert; 0 = Do not invert | bool | 1543 | 5443 | Not applicable |
| DigitalIO.DI_LBLC.MinOnTime | Time proportioned output minimum on time | float32 | 1542 | 5442 | 2dp |
| DigitalIO.DI_LBLC.ModuleIdent | As DigitalIO.1A1B.ModuleIdent | uint8 | 154a | 5450 | Not applicable |
| DigitalIO.DI_LBLC.Output | 0 = Output off, 1 = Output on | bool | 1544 | 5444 | Not applicable |
| DigitalIO.DI_LBLC.PV | Digital I/O process value (as DigitalIO.1A1B.PV) | float32 | 1541 | 5441 | 0dp |
| DigitalIO.DI_LBLC.StandbyAction | Valve positioning standby action (0 = Continue; 1 = Freeze). | uint8 | 1549 | 5449 | Not applicable |
| DigitalIO.DI_LBLC.Type | Digital I/O type (as DigitalIO.1A1B.Type). | uint8 | 1540 | 5440 | Not applicable |
| DigitalIO.RELAY_4AC.Backlash | Valve positioning backlash compensation (seconds) | float32 | 1558 | 5464 | 1dp |
| DigitalIO.RELAY_4AC.Inertia | Inertia value for the valve | float32 | 1557 | 5463 | 1dp |
| DigitalIO.RELAY_4AC.Invert | 1 = Invert; 0 = Do not invert | bool | 1553 | 5459 | Not applicable |
| DigitalIO.RELAY_4AC.MinOnTime | Time proportioned output minimum on time | float32 | 1552 | 5458 | 2dp |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| DigitalIO.RELAY_4AC.ModuleIdent | As DigitalIO.1A1B.ModuleIdent | uint8 | 155a | 5466 | Not applicable |
| DigitalIO.RELAY_4AC.Output | 0 = Output off, 1 = Output on | bool | 1554 | 5460 | Not applicable |
| DigitalIO.RELAY_4AC.PV | Digital I/O process value (as DigitalIO.1A1B.PV) | float32 | 1551 | 5457 | 0dp |
| DigitalIO.RELAY_4AC.StandbyAction | Valve positioning standby action (0 = Continue; 1 = Freeze). | uint8 | 1559 | 5465 | Not applicable |
| DigitalIO.RELAY_4AC.Type | Digital I/O type (as DigitalIO.1A1B.Type). | uint8 | 1550 | 5456 | Not applicable |
| DigitalIO.RELAY_5AC.Backlash | Valve positioning backlash compensation (seconds) | float32 | 1568 | 5480 | 1dp |
| DigitalIO.RELAY_5AC.Inertia | Inertia value for the valve | float32 | 1567 | 5479 | 1dp |
| DigitalIO.RELAY_5AC.Invert | 1 = Invert; 0 = Do not invert | bool | 1563 | 5475 | Not applicable |
| DigitalIO.RELAY_5AC.MinOnTime | Time proportioned output minimum on time | float32 | 1562 | 5474 | 2dp |
| DigitalIO.RELAY_5AC.ModuleIdent | As DigitalIO.1A1B.ModuleIdent | uint8 | 156a | 5482 | Not applicable |
| DigitalIO.RELAY_5AC.Output | 0 = Output off, 1 = Output on | bool | 1564 | 5476 | Not applicable |
| DigitalIO.RELAY_5AC.PV | Digital I/O process value (as DigitalIO.1A1B.PV) | float32 | 1561 | 5473 | 0dp |
| DigitalIO.RELAY_5AC.StandbyAction | Valve positioning standby action (0 = Continue; 1 = Freeze). | uint8 | 1569 | 5481 | Not applicable |
| DigitalIO.RELAY_5AC.Type | Digital I/O type (as DigitalIO.1A1B.Type). | uint8 | 1560 | 5472 | Not applicable |
| EthernetIP.ImplicitInputs.Input1 | Read only input from an EtherNet/IP client | eint32 | 7e66 | 32358 | Not applicable |
| EthernetIP.ImplicitInputs.Input2 | See input 1 for details | eint32 | 7e6a | 32362 | Not applicable |
| EthernetIP.ImplicitInputs.Input3 | See input 1 for details | eint32 | 7e6e | 32366 | Not applicable |
| EthernetIP.ImplicitInputs.Input4 | See input 1 for details | eint32 | 7e72 | 32370 | Not applicable |
| EthernetIP.ImplicitInputs.Input5 | See input 1 for details | eint32 | 7e76 | 32374 | Not applicable |
| EthernetIP.ImplicitInputs.Input6 | See input 1 for details | eint32 | 7e7a | 32378 | Not applicable |
| EthernetIP.ImplicitInputs.Input7 | See input 1 for details | eint32 | 7e7e | 32382 | Not applicable |
| EthernetIP.ImplicitInputs.Input8 | See input 1 for details | eint32 | 7e82 | 32386 | Not applicable |
| EthernetIP.ImplicitInputs.Input9 | See input 1 for details | eint32 | 7e86 | 32390 | Not applicable |
| EthernetIP.ImplicitInputs.Input10 | See input 1 for details | eint32 | 7e8a | 32394 | Not applicable |
| EthernetIP.ImplicitInputs.Input11 | See input 1 for details | eint32 | 7e8e | 32398 | Not applicable |
| EthernetIP.ImplicitInputs.Input12 | See input 1 for details | eint32 | 7e92 | 32402 | Not applicable |
| EthernetIP.ImplicitInputs.Input13 | See input 1 for details | eint32 | 7e96 | 32406 | Not applicable |
| EthernetIP.ImplicitInputs.Input14 | See input 1 for details | eint32 | 7e9a | 32410 | Not applicable |
| EthernetIP.ImplicitInputs.Input15 | See input 1 for details | eint32 | 7e9e | 32414 | Not applicable |
| EthernetIP.ImplicitInputs.Input16 | See input 1 for details | eint32 | 7ea2 | 32418 | Not applicable |
| EthernetIP.ImplicitInputs.Input17 | See input 1 for details | eint32 | 7ea6 | 32422 | Not applicable |
| EthernetIP.ImplicitInputs.Input18 | See input 1 for details | eint32 | 7eaa | 32426 | Not applicable |
| EthernetIP.ImplicitInputs.Input19 | See input 1 for details | eint32 | 7eae | 32430 | Not applicable |
| EthernetIP.ImplicitInputs.Input20 | See input 1 for details | eint32 | 7eb2 | 32434 | Not applicable |
| EthernetIP.ImplicitInputs.Input21 | See input 1 for details | eint32 | 7eb6 | 32438 | Not applicable |
| EthernetIP.ImplicitInputs.Input22 | See input 1 for details | eint32 | 7eba | 32442 | Not applicable |
| EthernetIP.ImplicitInputs.Input23 | See input 1 for details | eint32 | 7ebe | 32446 | Not applicable |
| EthernetIP.ImplicitInputs.Input24 | See input 1 for details | eint32 | 7ec2 | 32450 | Not applicable |
| EthernetIP.ImplicitInputs.Input25 | See input 1 for details | eint32 | 7ec6 | 32454 | Not applicable |
| EthernetIP.ImplicitInputs.Input26 | See input 1 for details | eint32 | 7eca | 32458 | Not applicable |
| EthernetIP.ImplicitInputs.Input27 | See input 1 for details | eint32 | 7ece | 32462 | Not applicable |
| EthernetIP.ImplicitInputs.Input28 | See input 1 for details | eint32 | 7ed2 | 32466 | Not applicable |
| EthernetIP.ImplicitInputs.Input29 | See input 1 for details | eint32 | 7ed6 | 32470 | Not applicable |
| EthernetIP.ImplicitInputs.Input30 | See input 1 for details | eint32 | 7eda | 32474 | Not applicable |
| EthernetIP.ImplicitInputs.Input31 | See input 1 for details | eint32 | 7ede | 32478 | Not applicable |
| EthernetIP.ImplicitInputs.Input32 | See input 1 for details | eint32 | 7ee2 | 32482 | Not applicable |
| EthernetIP.ImplicitInputs.Input33 | See input 1 for details | eint32 | 7ee6 | 32486 | Not applicable |
| EthernetIP.ImplicitInputs.Input34 | See input 1 for details | eint32 | 7eea | 32490 | Not applicable |
| EthernetIP.ImplicitInputs.Input35 | See input 1 for details | eint32 | 7eee | 32494 | Not applicable |
| EthernetIP.ImplicitInputs.Input36 | See input 1 for details | eint32 | 7ef2 | 32498 | Not applicable |
| EthernetIP.ImplicitInputs.Input37 | See input 1 for details | eint32 | 7ef6 | 32502 | Not applicable |
| EthernetIP.ImplicitInputs.Input38 | See input 1 for details | eint32 | 7efa | 32506 | Not applicable |
| EthernetIP.ImplicitInputs.Input39 | See input 1 for details | eint32 | 7efe | 32510 | Not applicable |
| EthernetIP.ImplicitInputs.Input40 | See input 1 for details | eint32 | 7f02 | 32514 | Not applicable |
| EthernetIP.ImplicitInputs.Input41 | See input 1 for details | eint32 | 7f06 | 32518 | Not applicable |
| EthernetIP.ImplicitInputs.Input42 | See input 1 for details | eint32 | 7f0a | 32522 | Not applicable |
| EthernetIP.ImplicitInputs.Input43 | See input 1 for details | eint32 | 7f0e | 32526 | Not applicable |
| EthernetIP.ImplicitInputs.Input44 | See input 1 for details | eint32 | 7f12 | 32530 | Not applicable |
| EthernetIP.ImplicitInputs.Input45 | See input 1 for details | eint32 | 7f16 | 32534 | Not applicable |
| EthernetIP.ImplicitInputs.Input46 | See input 1 for details | eint32 | 7f1a | 32538 | Not applicable |
| EthernetIP.ImplicitInputs.Input47 | See input 1 for details | eint32 | 7f1e | 32542 | Not applicable |
| EthernetIP.ImplicitInputs.Input48 | See input 1 for details | eint32 | 7f22 | 32546 | Not applicable |
| EthernetIP.ImplicitInputs.Input49 | See input 1 for details | eint32 | 7f26 | 32550 | Not applicable |
| EthernetIP.ImplicitInputs.Input50 | See input 1 for details | eint32 | 7f2a | 32554 | Not applicable |
| EthernetIP.ImplicitInputs.InputValue1 | Value of the Input 1 parameter | int16 | 7e68 | 32360 | Not applicable |
| EthernetIP.ImplicitInputs.InputValue2 | See input 1 value for details | int16 | 7e6c | 32364 | Not applicable |
| EthernetIP.ImplicitInputs.InputValue3 | See input 1 value for details | int16 | 7e70 | 32368 | Not applicable |
| EthernetIP.ImplicitInputs.InputValue4 | See input 1 value for details | int16 | 7e74 | 32372 | Not applicable |
| EthernetIP.ImplicitInputs.InputValue5 | See input 1 value for details | int16 | 7e78 | 32376 | Not applicable |
| EthernetIP.ImplicitInputs.InputValue6 | See input 1 value for details | int16 | 7e7c | 32380 | Not applicable |
| EthernetIP.ImplicitInputs.InputValue7 | See input 1 value for details | int16 | 7e80 | 32384 | Not applicable |
| EthernetIP.ImplicitInputs.InputValue8 | See input 1 value for details | int16 | 7e84 | 32388 | Not applicable |
| EthernetIP.ImplicitInputs.InputValue9 | See input 1 value for details | int16 | 7e88 | 32392 | Not applicable |
| EthernetIP.ImplicitInputs.InputValue10 | See input 1 value for details | int16 | 7e8c | 32396 | Not applicable |
| EthernetIP.ImplicitInputs.InputValue11 | See input 1 value for details | int16 | 7e90 | 32400 | Not applicable |
| EthernetIP.ImplicitInputs.InputValue12 | See input 1 value for details | int16 | 7e94 | 32404 | Not applicable |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| EthernetIP.ImplicitInputs.InputValue13 | See input 1 value for details | int16 | 7e98 | 32408 | Not applicable |
| EthernetIP.ImplicitInputs.InputValue14 | See input 1 value for details | int16 | 7e9c | 32412 | Not applicable |
| EthernetIP.ImplicitInputs.InputValue15 | See input 1 value for details | int16 | 7ea0 | 32416 | Not applicable |
| EthernetIP.ImplicitInputs.InputValue16 | See input 1 value for details | int16 | 7ea4 | 32420 | Not applicable |
| EthernetIP.ImplicitInputs.InputValue17 | See input 1 value for details | int16 | 7ea8 | 32424 | Not applicable |
| EthernetIP.ImplicitInputs.InputValue18 | See input 1 value for details | int16 | 7eac | 32428 | Not applicable |
| EthernetIP.ImplicitInputs.InputValue19 | See input 1 value for details | int16 | 7eb0 | 32432 | Not applicable |
| EthernetIP.ImplicitInputs.InputValue20 | See input 1 value for details | int16 | 7eb4 | 32436 | Not applicable |
| EthernetIP.ImplicitInputs.InputValue21 | See input 1 value for details | int16 | 7eb8 | 32440 | Not applicable |
| EthernetIP.ImplicitInputs.InputValue22 | See input 1 value for details | int16 | 7ebc | 32444 | Not applicable |
| EthernetIP.ImplicitInputs.InputValue23 | See input 1 value for details | int16 | 7ec0 | 32448 | Not applicable |
| EthernetIP.ImplicitInputs.InputValue24 | See input 1 value for details | int16 | 7ec4 | 32452 | Not applicable |
| EthernetIP.ImplicitInputs.InputValue25 | See input 1 value for details | int16 | 7ec8 | 32456 | Not applicable |
| EthernetIP.ImplicitInputs.InputValue26 | See input 1 value for details | int16 | 7ecc | 32460 | Not applicable |
| EthernetIP.ImplicitInputs.InputValue27 | See input 1 value for details | int16 | 7ed0 | 32464 | Not applicable |
| EthernetIP.ImplicitInputs.InputValue28 | See input 1 value for details | int16 | 7ed4 | 32468 | Not applicable |
| EthernetIP.ImplicitInputs.InputValue29 | See input 1 value for details | int16 | 7ed8 | 32472 | Not applicable |
| EthernetIP.ImplicitInputs.InputValue30 | See input 1 value for details | int16 | 7edc | 32476 | Not applicable |
| EthernetIP.ImplicitInputs.InputValue31 | See input 1 value for details | int16 | 7ee0 | 32480 | Not applicable |
| EthernetIP.ImplicitInputs.InputValue32 | See input 1 value for details | int16 | 7ee4 | 32484 | Not applicable |
| EthernetIP.ImplicitInputs.InputValue33 | See input 1 value for details | int16 | 7ee8 | 32488 | Not applicable |
| EthernetIP.ImplicitInputs.InputValue34 | See input 1 value for details | int16 | 7eec | 32492 | Not applicable |
| EthernetIP.ImplicitInputs.InputValue35 | See input 1 value for details | int16 | 7ef0 | 32496 | Not applicable |
| EthernetIP.ImplicitInputs.InputValue36 | See input 1 value for details | int16 | 7ef4 | 32500 | Not applicable |
| EthernetIP.ImplicitInputs.InputValue37 | See input 1 value for details | int16 | 7ef8 | 32504 | Not applicable |
| EthernetIP.ImplicitInputs.InputValue38 | See input 1 value for details | int16 | 7efc | 32508 | Not applicable |
| EthernetIP.ImplicitInputs.InputValue39 | See input 1 value for details | int16 | 7f00 | 32512 | Not applicable |
| EthernetIP.ImplicitInputs.InputValue40 | See input 1 value for details | int16 | 7f04 | 32516 | Not applicable |
| EthernetIP.ImplicitInputs.InputValue41 | See input 1 value for details | int16 | 7f08 | 32520 | Not applicable |
| EthernetIP.ImplicitInputs.InputValue42 | See input 1 value for details | int16 | 7f0c | 32524 | Not applicable |
| EthernetIP.ImplicitInputs.InputValue43 | See input 1 value for details | int16 | 7f10 | 32528 | Not applicable |
| EthernetIP.ImplicitInputs.InputValue44 | See input 1 value for details | int16 | 7f14 | 32532 | Not applicable |
| EthernetIP.ImplicitInputs.InputValue45 | See input 1 value for details | int16 | 7f18 | 32536 | Not applicable |
| EthernetIP.ImplicitInputs.InputValue46 | See input 1 value for details | int16 | 7f1c | 32540 | Not applicable |
| EthernetIP.ImplicitInputs.InputValue47 | See input 1 value for details | int16 | 7f20 | 32544 | Not applicable |
| EthernetIP.ImplicitInputs.InputValue48 | See input 1 value for details | int16 | 7f24 | 32548 | Not applicable |
| EthernetIP.ImplicitInputs.InputValue49 | See input 1 value for details | int16 | 7f28 | 32552 | Not applicable |
| EthernetIP.ImplicitInputs.InputValue50 | See input 1 value for details | int16 | 7f2c | 32556 | Not applicable |
| | | | | | |
| EthernetIP.ImplicitOutputs.Output1 | Writable output to the EtherNet/IP client | eint32 | 7f2e | 32558 | Not applicable |
| EthernetIP.ImplicitOutputs.Output2 | See output 1 for details | eint32 | 7f32 | 32562 | Not applicable |
| EthernetIP.ImplicitOutputs.Output3 | See output 1 for details | eint32 | 7f36 | 32566 | Not applicable |
| EthernetIP.ImplicitOutputs.Output4 | See output 1 for details | eint32 | 7f3a | 32570 | Not applicable |
| EthernetIP.ImplicitOutputs.Output5 | See output 1 for details | eint32 | 7f3e | 32574 | Not applicable |
| EthernetIP.ImplicitOutputs.Output6 | See output 1 for details | eint32 | 7f42 | 32578 | Not applicable |
| EthernetIP.ImplicitOutputs.Output7 | See output 1 for details | eint32 | 7f46 | 32582 | Not applicable |
| EthernetIP.ImplicitOutputs.Output8 | See output 1 for details | eint32 | 7f4a | 32586 | Not applicable |
| EthernetIP.ImplicitOutputs.Output9 | See output 1 for details | eint32 | 7f4e | 32590 | Not applicable |
| EthernetIP.ImplicitOutputs.Output10 | See output 1 for details | eint32 | 7f52 | 32594 | Not applicable |
| EthernetIP.ImplicitOutputs.Output11 | See output 1 for details | eint32 | 7f56 | 32598 | Not applicable |
| EthernetIP.ImplicitOutputs.Output12 | See output 1 for details | eint32 | 7f5a | 32602 | Not applicable |
| EthernetIP.ImplicitOutputs.Output13 | See output 1 for details | eint32 | 7f5e | 32606 | Not applicable |
| EthernetIP.ImplicitOutputs.Output14 | See output 1 for details | eint32 | 7f62 | 32610 | Not applicable |
| EthernetIP.ImplicitOutputs.Output15 | See output 1 for details | eint32 | 7f66 | 32614 | Not applicable |
| EthernetIP.ImplicitOutputs.Output16 | See output 1 for details | eint32 | 7f6a | 32618 | Not applicable |
| EthernetIP.ImplicitOutputs.Output17 | See output 1 for details | eint32 | 7f6e | 32622 | Not applicable |
| EthernetIP.ImplicitOutputs.Output18 | See output 1 for details | eint32 | 7f72 | 32626 | Not applicable |
| EthernetIP.ImplicitOutputs.Output19 | See output 1 for details | eint32 | 7f76 | 32630 | Not applicable |
| EthernetIP.ImplicitOutputs.Output20 | See output 1 for details | eint32 | 7f7a | 32634 | Not applicable |
| EthernetIP.ImplicitOutputs.Output21 | See output 1 for details | eint32 | 7f7e | 32638 | Not applicable |
| EthernetIP.ImplicitOutputs.Output22 | See output 1 for details | eint32 | 7f82 | 32642 | Not applicable |
| EthernetIP.ImplicitOutputs.Output23 | See output 1 for details | eint32 | 7f86 | 32646 | Not applicable |
| EthernetIP.ImplicitOutputs.Output24 | See output 1 for details | eint32 | 7f8a | 32650 | Not applicable |
| EthernetIP.ImplicitOutputs.Output25 | See output 1 for details | eint32 | 7f8e | 32654 | Not applicable |
| EthernetIP.ImplicitOutputs.Output26 | See output 1 for details | eint32 | 7f92 | 32658 | Not applicable |
| EthernetIP.ImplicitOutputs.Output27 | See output 1 for details | eint32 | 7f96 | 32662 | Not applicable |
| EthernetIP.ImplicitOutputs.Output28 | See output 1 for details | eint32 | 7f9a | 32666 | Not applicable |
| EthernetIP.ImplicitOutputs.Output29 | See output 1 for details | eint32 | 7f9e | 32670 | Not applicable |
| EthernetIP.ImplicitOutputs.Output30 | See output 1 for details | eint32 | 7fa2 | 32674 | Not applicable |
| EthernetIP.ImplicitOutputs.Output31 | See output 1 for details | eint32 | 7fa6 | 32678 | Not applicable |
| EthernetIP.ImplicitOutputs.Output32 | See output 1 for details | eint32 | 7faa | 32682 | Not applicable |
| EthernetIP.ImplicitOutputs.Output33 | See output 1 for details | eint32 | 7fae | 32686 | Not applicable |
| EthernetIP.ImplicitOutputs.Output34 | See output 1 for details | eint32 | 7fb2 | 32690 | Not applicable |
| EthernetIP.ImplicitOutputs.Output35 | See output 1 for details | eint32 | 7fb6 | 32694 | Not applicable |
| EthernetIP.ImplicitOutputs.Output36 | See output 1 for details | eint32 | 7fba | 32698 | Not applicable |
| EthernetIP.ImplicitOutputs.Output37 | See output 1 for details | eint32 | 7fbe | 32702 | Not applicable |
| EthernetIP.ImplicitOutputs.Output38 | See output 1 for details | eint32 | 7fc2 | 32706 | Not applicable |
| EthernetIP.ImplicitOutputs.Output39 | See output 1 for details | eint32 | 7fc6 | 32710 | Not applicable |
| EthernetIP.ImplicitOutputs.Output40 | See output 1 for details | eint32 | 7fca | 32714 | Not applicable |
| EthernetIP.ImplicitOutputs.Output41 | See output 1 for details | eint32 | 7fce | 32718 | Not applicable |
| EthernetIP.ImplicitOutputs.Output42 | See output 1 for details | eint32 | 7fd2 | 32722 | Not applicable |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| EthernetIP.ImplicitOutputs.Output43 | See output 1 for details | eint32 | 7fd6 | 32726 | Not applicable |
| EthernetIP.ImplicitOutputs.Output44 | See output 1 for details | eint32 | 7fda | 32730 | Not applicable |
| EthernetIP.ImplicitOutputs.Output45 | See output 1 for details | eint32 | 7fde | 32734 | Not applicable |
| EthernetIP.ImplicitOutputs.Output46 | See output 1 for details | eint32 | 7fe2 | 32738 | Not applicable |
| EthernetIP.ImplicitOutputs.Output47 | See output 1 for details | eint32 | 7fe6 | 32742 | Not applicable |
| EthernetIP.ImplicitOutputs.Output48 | See output 1 for details | eint32 | 7fea | 32746 | Not applicable |
| EthernetIP.ImplicitOutputs.Output49 | See output 1 for details | eint32 | 7fee | 32750 | Not applicable |
| EthernetIP.ImplicitOutputs.Output50 | See output 1 for details | eint32 | 7ff2 | 32754 | Not applicable |
| EthernetIP.ImplicitOutputs.OutputValue1 | Value of the Output 1 parameter | int16 | 7f30 | 32560 | Not applicable |
| EthernetIP.ImplicitOutputs.OutputValue2 | See output 1 value for details | int16 | 7f34 | 32564 | Not applicable |
| EthernetIP.ImplicitOutputs.OutputValue3 | See output 1 value for details | int16 | 7f38 | 32568 | Not applicable |
| EthernetIP.ImplicitOutputs.OutputValue4 | See output 1 value for details | int16 | 7f3c | 32572 | Not applicable |
| EthernetIP.ImplicitOutputs.OutputValue5 | See output 1 value for details | int16 | 7f40 | 32576 | Not applicable |
| EthernetIP.ImplicitOutputs.OutputValue6 | See output 1 value for details | int16 | 7f44 | 32580 | Not applicable |
| EthernetIP.ImplicitOutputs.OutputValue7 | See output 1 value for details | int16 | 7f48 | 32584 | Not applicable |
| EthernetIP.ImplicitOutputs.OutputValue8 | See output 1 value for details | int16 | 7f4c | 32588 | Not applicable |
| EthernetIP.ImplicitOutputs.OutputValue9 | See output 1 value for details | int16 | 7f50 | 32592 | Not applicable |
| EthernetIP.ImplicitOutputs.OutputValue10 | See output 1 value for details | int16 | 7f54 | 32596 | Not applicable |
| EthernetIP.ImplicitOutputs.OutputValue11 | See output 1 value for details | int16 | 7f58 | 32600 | Not applicable |
| EthernetIP.ImplicitOutputs.OutputValue12 | See output 1 value for details | int16 | 7f5c | 32604 | Not applicable |
| EthernetIP.ImplicitOutputs.OutputValue13 | See output 1 value for details | int16 | 7f60 | 32608 | Not applicable |
| EthernetIP.ImplicitOutputs.OutputValue14 | See output 1 value for details | int16 | 7f64 | 32612 | Not applicable |
| EthernetIP.ImplicitOutputs.OutputValue15 | See output 1 value for details | int16 | 7f68 | 32616 | Not applicable |
| EthernetIP.ImplicitOutputs.OutputValue16 | See output 1 value for details | int16 | 7f6c | 32620 | Not applicable |
| EthernetIP.ImplicitOutputs.OutputValue17 | See output 1 value for details | int16 | 7f70 | 32624 | Not applicable |
| EthernetIP.ImplicitOutputs.OutputValue18 | See output 1 value for details | int16 | 7f74 | 32628 | Not applicable |
| EthernetIP.ImplicitOutputs.OutputValue19 | See output 1 value for details | int16 | 7f78 | 32632 | Not applicable |
| EthernetIP.ImplicitOutputs.OutputValue20 | See output 1 value for details | int16 | 7f7c | 32636 | Not applicable |
| EthernetIP.ImplicitOutputs.OutputValue21 | See output 1 value for details | int16 | 7f80 | 32640 | Not applicable |
| EthernetIP.ImplicitOutputs.OutputValue22 | See output 1 value for details | int16 | 7f84 | 32644 | Not applicable |
| EthernetIP.ImplicitOutputs.OutputValue23 | See output 1 value for details | int16 | 7f88 | 32648 | Not applicable |
| EthernetIP.ImplicitOutputs.OutputValue24 | See output 1 value for details | int16 | 7f8c | 32652 | Not applicable |
| EthernetIP.ImplicitOutputs.OutputValue25 | See output 1 value for details | int16 | 7f90 | 32656 | Not applicable |
| EthernetIP.ImplicitOutputs.OutputValue26 | See output 1 value for details | int16 | 7f94 | 32660 | Not applicable |
| EthernetIP.ImplicitOutputs.OutputValue27 | See output 1 value for details | int16 | 7f98 | 32664 | Not applicable |
| EthernetIP.ImplicitOutputs.OutputValue28 | See output 1 value for details | int16 | 7f9c | 32668 | Not applicable |
| EthernetIP.ImplicitOutputs.OutputValue29 | See output 1 value for details | int16 | 7fa0 | 32672 | Not applicable |
| EthernetIP.ImplicitOutputs.OutputValue30 | See output 1 value for details | int16 | 7fa4 | 32676 | Not applicable |
| EthernetIP.ImplicitOutputs.OutputValue31 | See output 1 value for details | int16 | 7fa8 | 32680 | Not applicable |
| EthernetIP.ImplicitOutputs.OutputValue32 | See output 1 value for details | int16 | 7fac | 32684 | Not applicable |
| EthernetIP.ImplicitOutputs.OutputValue33 | See output 1 value for details | int16 | 7fb0 | 32688 | Not applicable |
| EthernetIP.ImplicitOutputs.OutputValue34 | See output 1 value for details | int16 | 7fb4 | 32692 | Not applicable |
| EthernetIP.ImplicitOutputs.OutputValue35 | See output 1 value for details | int16 | 7fb8 | 32696 | Not applicable |
| EthernetIP.ImplicitOutputs.OutputValue36 | See output 1 value for details | int16 | 7fbc | 32700 | Not applicable |
| EthernetIP.ImplicitOutputs.OutputValue37 | See output 1 value for details | int16 | 7fc0 | 32704 | Not applicable |
| EthernetIP.ImplicitOutputs.OutputValue38 | See output 1 value for details | int16 | 7fc4 | 32708 | Not applicable |
| EthernetIP.ImplicitOutputs.OutputValue39 | See output 1 value for details | int16 | 7fc8 | 32712 | Not applicable |
| EthernetIP.ImplicitOutputs.OutputValue40 | See output 1 value for details | int16 | 7fcc | 32716 | Not applicable |
| EthernetIP.ImplicitOutputs.OutputValue41 | See output 1 value for details | int16 | 7fd0 | 32720 | Not applicable |
| EthernetIP.ImplicitOutputs.OutputValue42 | See output 1 value for details | int16 | 7fd4 | 32724 | Not applicable |
| EthernetIP.ImplicitOutputs.OutputValue43 | See output 1 value for details | int16 | 7fd8 | 32728 | Not applicable |
| EthernetIP.ImplicitOutputs.OutputValue44 | See output 1 value for details | int16 | 7fdc | 32732 | Not applicable |
| EthernetIP.ImplicitOutputs.OutputValue45 | See output 1 value for details | int16 | 7fe0 | 32736 | Not applicable |
| EthernetIP.ImplicitOutputs.OutputValue46 | See output 1 value for details | int16 | 7fe4 | 32740 | Not applicable |
| EthernetIP.ImplicitOutputs.OutputValue47 | See output 1 value for details | int16 | 7fe8 | 32744 | Not applicable |
| EthernetIP.ImplicitOutputs.OutputValue48 | See output 1 value for details | int16 | 7fec | 32748 | Not applicable |
| EthernetIP.ImplicitOutputs.OutputValue49 | See output 1 value for details | int16 | 7ff0 | 32752 | Not applicable |
| EthernetIP.ImplicitOutputs.OutputValue50 | See output 1 value for details | int16 | 7ff4 | 32756 | Not applicable |
| EthernetIP.InputTags.Input1 | A read only input from a PLC device | string_t | 7838 | 30776 | Not applicable |
| EthernetIP.InputTags.Input2 | See input 1 for details | string_t | 7839 | 30777 | Not applicable |
| EthernetIP.InputTags.Input3 | See input 1 for details | string_t | 783a | 30778 | Not applicable |
| EthernetIP.InputTags.Input4 | See input 1 for details | string_t | 783b | 30779 | Not applicable |
| EthernetIP.InputTags.Input5 | See input 1 for details | string_t | 783c | 30780 | Not applicable |
| EthernetIP.InputTags.Input6 | See input 1 for details | string_t | 783d | 30781 | Not applicable |
| EthernetIP.InputTags.Input7 | See input 1 for details | string_t | 783e | 30782 | Not applicable |
| EthernetIP.InputTags.Input8 | See input 1 for details | string_t | 783f | 30783 | Not applicable |
| EthernetIP.InputTags.Input9 | See input 1 for details | string_t | 7840 | 30784 | Not applicable |
| EthernetIP.InputTags.Input10 | See input 1 for details | string_t | 7841 | 30785 | Not applicable |
| EthernetIP.InputTags.Input11 | See input 1 for details | string_t | 7842 | 30786 | Not applicable |
| EthernetIP.InputTags.Input12 | See input 1 for details | string_t | 7843 | 30787 | Not applicable |
| EthernetIP.InputTags.Input13 | See input 1 for details | string_t | 7844 | 30788 | Not applicable |
| EthernetIP.InputTags.Input14 | See input 1 for details | string_t | 7845 | 30789 | Not applicable |
| EthernetIP.InputTags.Input15 | See input 1 for details | string_t | 7846 | 30790 | Not applicable |
| EthernetIP.InputTags.Input16 | See input 1 for details | string_t | 7847 | 30791 | Not applicable |
| EthernetIP.InputTags.Input17 | See input 1 for details | string_t | 7848 | 30792 | Not applicable |
| EthernetIP.InputTags.Input18 | See input 1 for details | string_t | 7849 | 30793 | Not applicable |
| EthernetIP.InputTags.Input19 | See input 1 for details | string_t | 784a | 30794 | Not applicable |
| EthernetIP.InputTags.Input20 | See input 1 for details | string_t | 784b | 30795 | Not applicable |
| EthernetIP.InputTags.Input21 | See input 1 for details | string_t | 784c | 30796 | Not applicable |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| EthernetIP.InputTags.Input22 | See input 1 for details | string_t | 784d | 30797 | Not applicable |
| EthernetIP.InputTags.Input23 | See input 1 for details | string_t | 784e | 30798 | Not applicable |
| EthernetIP.InputTags.Input24 | See input 1 for details | string_t | 784f | 30799 | Not applicable |
| EthernetIP.InputTags.Input25 | See input 1 for details | string_t | 7850 | 30800 | Not applicable |
| EthernetIP.InputTags.Input26 | See input 1 for details | string_t | 7851 | 30801 | Not applicable |
| EthernetIP.InputTags.Input27 | See input 1 for details | string_t | 7852 | 30802 | Not applicable |
| EthernetIP.InputTags.Input28 | See input 1 for details | string_t | 7853 | 30803 | Not applicable |
| EthernetIP.InputTags.Input29 | See input 1 for details | string_t | 7854 | 30804 | Not applicable |
| EthernetIP.InputTags.Input30 | See input 1 for details | string_t | 7855 | 30805 | Not applicable |
| EthernetIP.Main.ConfigInstance | Configuration assembly instance number | int16 | 7ffa | 32762 | Not applicable |
| EthernetIP.Main.ConfigSize | Configuration assembly data size in bytes | int16 | 7ffb | 32763 | Not applicable |
| EthernetIP.Main.ConnectionType | Implicit I/O connection type (0 = Point to point; 1 = Multicast) | uint8 | 7ffe | 32766 | Not applicable |
| EthernetIP.Main.Explicit1 | Explicit TCP connection 1 | string_t | 65f1 | 26097 | Not applicable |
| EthernetIP.Main.Explicit2 | Explicit TCP connection 2 | string_t | 6601 | 26113 | Not applicable |
| EthernetIP.Main.ImplicitIO | Implicit I/O data channel | string_t | 65e1 | 26081 | Not applicable |
| EthernetIP.Main.InputInstance | Implicit input assembly instance number | int16 | 7ff6 | 32758 | Not applicable |
| EthernetIP.Main.InputSize | Implicit input assembly data size in bytes | int16 | 7ff7 | 32759 | Not applicable |
| EthernetIP.Main.Mode | EtherNet/IP operation mode<br>0 = Server 1 = Client (IO) 2 = Client (Tags) | uint8 | 7fff | 32767 | Not applicable |
| EthernetIP.Main.Multicast | Implicit I/O data channel multicast address | string_t | 6611 | 26129 | Not applicable |
| EthernetIP.Main.NetworkStatusCode | EtherNet/IP communications network status<br>0 = Offline 2 = On line 3 = Connection timout<br>4 = Duplicate IP address 5 = Initialisation | uint8 | 7e64 | 32356 | Not applicable |
| EthernetIP.Main.OutputInstance | Implicit output assembly instance number | int16 | 7ff8 | 32760 | Not applicable |
| EthernetIP.Main.OutputSize | Implicit output assembly data size in bytes | int16 | 7ff9 | 32761 | Not applicable |
| EthernetIP.Main.Priority | Level of message priority<br>0 = Low 1 = High 2 = Scheduled 3 = Urgent | uint8 | 7ffc | 32764 | Not applicable |
| EthernetIP.Main.ResetComms | Resets the client or server communications (0 = No; 1 = Yes) | uint8 | 7e63 | 32355 | Not applicable |
| EthernetIP.Main.Rpi | Requested Packet Interval (milliseconds) | int16 | 7ffd | 32765 | Not applicable |
| EthernetIP.Main.ServerAddress | IP address of a server device | string_t | 7129 | 28969 | Not applicable |
| EthernetIP.Main.SlotNumber | PLC slot number | int16 | 7e60 | 32352 | Not applicable |
| EthernetIP.Main.TagStatusCode | EtherNet/IP Tag server status code (see table 4.10.1) | uint8 | 7e62 | 32354 | Not applicable |
| EthernetIP.Main.UCMM | Unconnected Message Manager (UCMM) | string_t | 65d1 | 26065 | Not applicable |
| EthernetIP.OutputTags.Output1 | Writable output to the PLC device | string_t | 7880 | 30848 | Not applicable |
| EthernetIP.OutputTags.Output2 | See output 1 for details | string_t | 7881 | 30849 | Not applicable |
| EthernetIP.OutputTags.Output3 | See output 1 for details | string_t | 7882 | 30850 | Not applicable |
| EthernetIP.OutputTags.Output4 | See output 1 for details | string_t | 7883 | 30851 | Not applicable |
| EthernetIP.OutputTags.Output5 | See output 1 for details | string_t | 7884 | 30852 | Not applicable |
| EthernetIP.OutputTags.Output6 | See output 1 for details | string_t | 7885 | 30853 | Not applicable |
| EthernetIP.OutputTags.Output7 | See output 1 for details | string_t | 7886 | 30854 | Not applicable |
| EthernetIP.OutputTags.Output8 | See output 1 for details | string_t | 7887 | 30855 | Not applicable |
| EthernetIP.OutputTags.Output9 | See output 1 for details | string_t | 7888 | 30856 | Not applicable |
| EthernetIP.OutputTags.Output10 | See output 1 for details | string_t | 7889 | 30857 | Not applicable |
| EthernetIP.OutputTags.Output11 | See output 1 for details | string_t | 788a | 30858 | Not applicable |
| EthernetIP.OutputTags.Output12 | See output 1 for details | string_t | 788b | 30859 | Not applicable |
| EthernetIP.OutputTags.Output13 | See output 1 for details | string_t | 788c | 30860 | Not applicable |
| EthernetIP.OutputTags.Output14 | See output 1 for details | string_t | 788d | 30861 | Not applicable |
| EthernetIP.OutputTags.Output15 | See output 1 for details | string_t | 788e | 30862 | Not applicable |
| EthernetIP.OutputTags.Output16 | See output 1 for details | string_t | 788f | 30863 | Not applicable |
| EthernetIP.OutputTags.Output17 | See output 1 for details | string_t | 7890 | 30864 | Not applicable |
| EthernetIP.OutputTags.Output18 | See output 1 for details | string_t | 7891 | 30865 | Not applicable |
| EthernetIP.OutputTags.Output19 | See output 1 for details | string_t | 7892 | 30866 | Not applicable |
| EthernetIP.OutputTags.Output20 | See output 1 for details | string_t | 7893 | 30867 | Not applicable |
| EthernetIP.OutputTags.Output21 | See output 1 for details | string_t | 7894 | 30868 | Not applicable |
| EthernetIP.OutputTags.Output22 | See output 1 for details | string_t | 7895 | 30869 | Not applicable |
| EthernetIP.OutputTags.Output23 | See output 1 for details | string_t | 7896 | 30870 | Not applicable |
| EthernetIP.OutputTags.Output24 | See output 1 for details | string_t | 7897 | 30871 | Not applicable |
| EthernetIP.OutputTags.Output25 | See output 1 for details | string_t | 7898 | 30872 | Not applicable |
| EthernetIP.OutputTags.Output26 | See output 1 for details | string_t | 7899 | 30873 | Not applicable |
| EthernetIP.OutputTags.Output27 | See output 1 for details | string_t | 789a | 30874 | Not applicable |
| EthernetIP.OutputTags.Output28 | See output 1 for details | string_t | 789b | 30875 | Not applicable |
| EthernetIP.OutputTags.Output29 | See output 1 for details | string_t | 789c | 30876 | Not applicable |
| EthernetIP.OutputTags.Output30 | See output 1 for details | string_t | 789d | 30877 | Not applicable |
| Group.Recording.Channel1En | Channel 1 enable (0 = Disabled; 1 = Enabled) | bool | 1023 | 4131 | Not applicable |
| Group.Recording.Channel2En | Channel 2 enable (0 = Disabled; 1 = Enabled) | bool | 1024 | 4132 | Not applicable |
| Group.Recording.Channel3En | Channel 3 enable (0 = Disabled; 1 = Enabled) | bool | 1025 | 4133 | Not applicable |
| Group.Recording.Channel4En | Channel 4 enable (0 = Disabled; 1 = Enabled) | bool | 1026 | 4134 | Not applicable |
| Group.Recording.Compression | The UHH file compression rate (0 = Normal; 1 = High) | uint8 | 1040 | 4160 | Not applicable |
| Group.Recording.Enable | 0 = Recording disabled; 1 = Recording enabled | uint8 | 1020 | 4128 | Not applicable |
| Group.Recording.FlashDuration | Time in days until flash history files begin to be overwitten | float32 | 1039 | 4153 | 2dp |
| Group.Recording.FlashFree | Size of the internal flash in MBytes | float32 | 1038 | 4152 | 2dp |
| Group.Recording.FlashSize | Size of the internal flash in MBytes | float32 | 1037 | 4151 | 2dp |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| Group.Recording.Interval | Recording interval<br>0 = 125 secs 1 = .25 secs 2 = 0.5 secs<br>3 = 1Hz 4 = 2 sec 5 = 5 sec<br>6 = 10 sec 7 = 20 sec 8 = 30 sec<br>9 = 1 min 10 = 2 min 11 = 5 min<br>12 = 10 min 13 = 20 min 14 = 30 min<br>15 = 1 hr | int32 | 1022 | 4130 | Not applicable |
| Group.Recording.Status | Recording status<br>0 = Not recording 1 = Disabled<br>2 = Messages only 3 = Recording enabled<br>4 = Recording paused | int16 | 1036 | 4150 | Not applicable |
| Group.Recording.Suspend | 1 = Suspend recording | bool | 1035 | 4149 | Not applicable |
| Group.Recording.VirtualChan1En | Virtual Channel 1 enable (0 = Disabled; 1 = Enabled) | bool | 1027 | 4135 | Not applicable |
| Group.Recording.VirtualChan2En | Virtual Channel 2 enable (0 = Disabled; 1 = Enabled) | bool | 1028 | 4136 | Not applicable |
| Group.Recording.VirtualChan3En | Virtual Channel 3 enable (0 = Disabled; 1 = Enabled) | bool | 1029 | 4137 | Not applicable |
| Group.Recording.VirtualChan4En | Virtual Channel 4 enable (0 = Disabled; 1 = Enabled) | bool | 102a | 4138 | Not applicable |
| Group.Recording.VirtualChan5En | Virtual Channel 5 enable (0 = Disabled; 1 = Enabled) | bool | 102b | 4139 | Not applicable |
| Group.Recording.VirtualChan6En | Virtual Channel 6 enable (0 = Disabled; 1 = Enabled) | bool | 102c | 4140 | Not applicable |
| Group.Recording.VirtualChan7En | Virtual Channel 7 enable (0 = Disabled; 1 = Enabled) | bool | 102d | 4141 | Not applicable |
| Group.Recording.VirtualChan8En | Virtual Channel 8 enable (0 = Disabled; 1 = Enabled) | bool | 102e | 4142 | Not applicable |
| Group.Recording.VirtualChan9En | Virtual Channel 9 enable (0 = Disabled; 1 = Enabled) | bool | 102f | 4143 | Not applicable |
| Group.Recording.VirtualChan10En | Virtual Channel 10 enable (0 = Disabled; 1 = Enabled) | bool | 1030 | 4144 | Not applicable |
| Group.Recording.VirtualChan11En | Virtual Channel 11 enable (0 = Disabled; 1 = Enabled) | bool | 1031 | 4145 | Not applicable |
| Group.Recording.VirtualChan12En | Virtual Channel 12 enable (0 = Disabled; 1 = Enabled) | bool | 1032 | 4146 | Not applicable |
| Group.Recording.VirtualChan13En | Virtual Channel 13 enable (0 = Disabled; 1 = Enabled) | bool | 1033 | 4147 | Not applicable |
| Group.Recording.VirtualChan14En | Virtual Channel 14 enable (0 = Disabled; 1 = Enabled) | bool | 1034 | 4148 | Not applicable |
| Group.Recording.VirtualChan15En | Virtual Channel 15 enable (0 = Disabled; 1 = Enabled) | bool | 103a | 4154 | Not applicable |
| Group.Recording.VirtualChan16En | Virtual Channel 16 enable (0 = Disabled; 1 = Enabled) | bool | 103b | 4155 | Not applicable |
| Group.Recording.VirtualChan17En | Virtual Channel 17 enable (0 = Disabled; 1 = Enabled) | bool | 103c | 4156 | Not applicable |
| Group.Recording.VirtualChan18En | Virtual Channel 18 enable (0 = Disabled; 1 = Enabled) | bool | 103d | 4157 | Not applicable |
| Group.Recording.VirtualChan19En | Virtual Channel 19 enable (0 = Disabled; 1 = Enabled) | bool | 103e | 4158 | Not applicable |
| Group.Recording.VirtualChan20En | Virtual Channel 20 enable (0 = Disabled; 1 = Enabled) | bool | 103f | 4159 | Not applicable |
| Group.Recording.VirtualChan21En | Virtual Channel 21 enable (0 = Disabled; 1 = Enabled) | bool | 1041 | 4161 | Not applicable |
| Group.Recording.VirtualChan22En | Virtual Channel 22 enable (0 = Disabled; 1 = Enabled) | bool | 1042 | 4162 | Not applicable |
| Group.Recording.VirtualChan23En | Virtual Channel 23 enable (0 = Disabled; 1 = Enabled) | bool | 1043 | 4163 | Not applicable |
| Group.Recording.VirtualChan24En | Virtual Channel 24 enable (0 = Disabled; 1 = Enabled) | bool | 1044 | 4164 | Not applicable |
| Group.Recording.VirtualChan25En | Virtual Channel 25 enable (0 = Disabled; 1 = Enabled) | bool | 1045 | 4165 | Not applicable |
| Group.Recording.VirtualChan26En | Virtual Channel 26 enable (0 = Disabled; 1 = Enabled) | bool | 1046 | 4166 | Not applicable |
| Group.Recording.VirtualChan27En | Virtual Channel 27 enable (0 = Disabled; 1 = Enabled) | bool | 1047 | 4167 | Not applicable |
| Group.Recording.VirtualChan28En | Virtual Channel 28 enable (0 = Disabled; 1 = Enabled) | bool | 1048 | 4168 | Not applicable |
| Group.Recording.VirtualChan29En | Virtual Channel 29 enable (0 = Disabled; 1 = Enabled) | bool | 1049 | 4169 | Not applicable |
| Group.Recording.VirtualChan30En | Virtual Channel 30 enable (0 = Disabled; 1 = Enabled) | bool | 104a | 4170 | Not applicable |
| Group.Trend.Descriptor | Group descriptor | string_t | 5b00 | 23296 | Not applicable |
| Group.Trend.Interval | Trend interval. As Group.Recording.Interval, above | int32 | 1002 | 4098 | Not applicable |
| Group.Trend.MajorDivisions | Number of major divisions | uint8 | 1004 | 4100 | Not applicable |
| Group.Trend.Point1 | 1st point in the group (VCh = Virtual channel)uint8<br>0 =No trend 1 = Channel 1 2 = Channel 2<br>3 = Channel 3 4 = Channel 4 5 = VCh1<br>6 = VCh2 7 = VCh3 8 = VCh4<br>9 = VCh5 10 = VCh6 11 = VCh7<br>12 = VCh8 13 = VCh9 14 = VCh10<br>15 = VCh11 16 = VCh12 17 = VCH13<br>18 = VCh14 19 = VCh15 20 = VCh 16<br>21 = VCh17 22 = VCh18 23 = VCh 19<br>24 = VCh20 25 = VCh21 26 = VCh 22<br>27 = VCh23 28 = VCh24 29 = VCh 25<br>30 = VCh26 31 = VCh27 32 = VCh 28<br>33 = VCh29 34 = VCh30 | 1006 | | 4102 | Not applicable |
| Group.Trend.Point2 | As Group.Trend.Point1 but for 2nd point in group | uint8 | 1007 | 4103 | Not applicable |
| Group.Trend.Point3 | As Group.Trend.Point1 but for 3rd point in group | uint8 | 1008 | 4104 | Not applicable |
| Group.Trend.Point4 | As Group.Trend.Point1 but for 4th point in group | uint8 | 1009 | 4105 | Not applicable |
| Group.Trend.Point5 | As Group.Trend.Point1 but for 5th point in group | uint8 | 100a | 4106 | Not applicable |
| Group.Trend.Point6 | As Group.Trend.Point1 but for 6th point in group | uint8 | 100b | 4107 | Not applicable |
| IHumidity.DewPoint | Dewpoint | float32 | 2e79 | 11897 | Set by Humidity.Resolution |
| Humidity.DryTemp | Dry Bulb Temperature Measurement | float32 | 2e7d | 11901 | 0dp |
| Humidity.Pressure | Current Atmospheric Pressure | float32 | 2e80 | 11904 | 1dp |
| Humidity.PsychroConst | Psychrometric Constant | float32 | 2e7f | 11903 | 2dp |
| Humidity.RelHumid | Calculated Relative Humidity | float32 | 2e78 | 11896 | Set by Humidity.Resolution |
| Humidity.Resolution | Result Resolution | uint8 | 2e81 | 11905 | Not applicable |
| Humidity.SBrk | Sensor Broken (0 = No; 1 = Yes) | bool | 2e7e | 11902 | Not applicable |
| Humidity.WetOffset | Offset of the Wet Bulb Temperature | float32 | 2e7b | 11899 | Same as Humidity.WetTemp |
| Humidity.WetTemp | Wet Bulb Temperature Measurement | float32 | 2e7c | 11900 | 0dp |
| Instrument.Clock.Date | Local Date | string_t | 4400 | 17408 | Not applicable |
| Instrument.Clock.DST | 1 = DST active; 0 = DST not active | bool | 1082 | 4226 | Not applicable |
| Instrument.Clock.Time | Local time (including Zone and DST effects) | time_t | 1081 | 4225 | Set by Network.Modbus.TimeFormat |
| Instrument.Display.AlarmPanel | 1 = Alarm Panel display mode enabled | bool | 10eb | 4331 | Not applicable |
| Instrument.Display.Brightness | Display brightness 10 = 10%; 20 = 20% etc. (whole decades) | uint8 | 1090 | 4240 | Not applicable |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| Instrument.Display.Cascade | 1 = Cascade control display mode enabled | bool | 10f2 | 4338 | Not applicable |
| Instrument.Display.DualLoopControl | 1 = Dual loop control display mode enabled | bool | 109b | 4251 | Not applicable |
| Instrument.Display.EIPServerPage | 1 = EtherNet/IPdisplay mode enabled | bool | 10ef | 4335 | Not applicable |
| Instrument.Display.FaceplateCycling | 1 = Faceplate cycling On | bool | 109e | 4254 | Not applicable |
| Instrument.Display.FutureTrend | 1 = Future trend display mode enabled | bool | 10fb | 4347 | Not applicable |
| Instrument.Display.FutureTrend1Colour | Future trend colour(1) (As Channel.1.Trend.Colour) | uint8 | 10fc | 4348 | Not applicable |
| Instrument.Display.FutureTrend2Colour | Future trend colour(2) (As Channel.1.Trend.Colour) | uint8 | 10fd | 4349 | Not applicable |
| Instrument.Display.HistoryBackground | History background colour<br>0 = Black; 1 = Dark grey; 2 = Light grey; 3 = White | uint8 | 10a8 | 4264 | Not applicable |
| Instrument.Display.HomePage | Home page | uint8 | 1093 | 4243 | Not applicable |
| Instrument.Display.HorizontalBar | 1 = Horizontal bar mode enabled | bool | 1098 | 4248 | Not applicable |
| Instrument.Display.HorizontalTrend | 1 = Horizontal trend mode enabled | bool | 1096 | 4246 | Not applicable |
| Instrument.Display.HPageTimeout | Home time out value in minutes (0 = no timeout) | int16 | 1094 | 4244 | Not applicable |
| Instrument.Display.HTrendScaling | 0 = hide horizontal trend scale; 1 = scale permanent | uint8 | 109d | 4253 | Not applicable |
| Instrument.Display.LoopControl | 1 = Loop control display mode enabled | bool | 109a | 4250 | Not applicable |
| Instrument.Display.LoopSetpointColour | Loop setpoint colour (As Channel.1.Trend.Colour) | uint8 | 109f | 4255 | Not applicable |
| Instrument.Display.ModbusMaster | 1 = Modbus Master display mode enabled | bool | 10ee | 4334 | Not applicable |
| Instrument.Display.NumberFormat | Number format (0 = Rounded; 1 - Truncated) | uint8 | 10fe | 4350 | Not applicable |
| Instrument.Display.Numeric | 1 = Numeric display mode enabled | bool | 1099 | 4249 | Not applicable |
| Instrument.Display.Programmer | 1 = Programmer interface display mode enabled | bool | 10f3 | 4339 | Not applicable |
| Instrument.Display.PromoteListView | 1 = Promote list display mode enabled | bool | 10ea | 4330 | Not applicable |
| Instrument.Display.ScreenSaverAfter | Screen save after (in minutes) | int16 | 1091 | 4241 | Not applicable |
| Instrument.Display.ScreenSaverBrightness | Screen saver brightness 10 = 10%; 20 = 20% etc.<br>(whole decades only) | uint8 | 1092 | 4242 | Not applicable |
| Instrument.Display.SteriliserPage | 1 = Steriliser display mode enabled | bool | 10ec | 4332 | Not applicable |
| Instrument.Display.TrendBackground | Trend chart colour:<br>0 = Black; 1 = Dark Grey; 2 = Light grey; 3 = White. | uint8 | 109c | 4252 | Not applicable |
| Instrument.Display.VerticalBar | 1 = Vertical bar display mode enabled | bool | 1097 | 4247 | Not applicable |
| Instrument.Display.VerticalTrend | 1 = Vertical trend display mode enabled | bool | 1095 | 4245 | Not applicable |
| Instrument.Info.Bootrom | Instrument bootrom version | string_t | 447a | 17530 | Not applicable |
| Instrument.Info.CompanyID | Company identification. Always returns 1280 | int16 | 0079 | 121 | Not applicable |
| Instrument.Info.ConfigRev | The instrument configuration revision number | int32 | 10a0 | 4256 | Not applicable |
| Instrument.Info.IM | Instrument mode<br>Operating: All algorithms and I/O active.<br>Standby: Control o/p off. Absolute alarms active<br>Engineer: All outputs inactive. | uint8 | 00c7 | 199 | Not applicable |
| Instrument.Info.LineVoltage | Displays the current line voltage | float32 | 10a6 | 4262 | 1dp |
| Instrument.Info.MicroBoardIssue | Micro Board Issue | uint8 | 10aa | 4266 | Not applicable |
| Instrument.Info.Name | The instrument descriptor | string_t | 445f | 17503 | Not applicable |
| Instrument.Info.NvolWrites | Displays the number of non-volatile writes performed | int32 | 10a5 | 4261 | Not applicable |
| Instrument.Info.PSUType | PSU type. 0 = 240Vac; 1 = 24v ac/dc | uint8 | 10a9 | 4265 | Not applicable |
| Instrument.Info.SecurityRev | The instrument security revision number | int32 | 10a4 | 4260 | Not applicable |
| Instrument.Info.Type | Instrument type | uint8 | 10a2 | 4258 | Not applicable |
| Instrument.Info.Version | Instrument version | string_t | 4474 | 17524 | Not applicable |
| Instrument.Info.WiresFree | Number of wires free | int16 | 10ab | 4267 | Not applicable |
| Instrument.IOFitted.1A1B | I/O fitted at terminals 1A1Buint8<br>0 = Digital IO 1= Non-isolated dc op (mA only)<br>2 = Relay op 3 = TRIAC 1a1b<br>4 = Relay OP 5 = Isolated dc op (V/mA)<br>6 = Digital ip 7 = Isolated dc output (mA only)<br>8 = Digital op 9 = Relay op<br>10 = Triac 2A2B | 10f4 | | 4340 | Not applicable |
| Instrument.IOFitted.2A2B | I/O fitted at terminals 2A2B (as for 1A1B above) | uint8 | 10f5 | 4341 | Not applicable |
| Instrument.IOFitted.3A3B | I/O type fitted at terminals 3A3B (as for 1A1B above) | uint8 | 10f7 | 4343 | Not applicable |
| Instrument.IOFitted.4AC | I/O type fitted at terminals 4AC (as for 1A1B above) | uint8 | 10f9 | 4345 | Not applicable |
| Instrument.IOFitted.5AC | I/O type fitted at terminals 5AC (as for 1A1B above) | uint8 | 10fa | 4346 | Not applicable |
| Instrument.IOFitted.LALC | I/O type fitted at terminals LALC (as for 1A1B above) | uint8 | 10f6 | 4342 | Not applicable |
| Instrument.IOFitted.LBLC | I/O type fitted at terminals LBLC (as for 1A1B above) | uint8 | 10f8 | 4344 | Not applicable |
| Instrument.Locale.DateFormat | Date format (0 = DDMMYY. 1 = MMDDYY; 2 = YYMMDD) | uint8 | 10b1 | 4273 | Not applicable |
| Instrument.Locale.DSTenable | 1 = Daylight Saving Time enabled | bool | 10b3 | 4275 | Not applicable |
| Instrument.Locale.EndDay | Daylight savings: End day<br>0 = Sunday 1= Monday 2 = Tuesday<br>3 = Wednesday 4 = Thursday 5 = Friday<br>6 = Saturday | uint8 | 10ba | 4282 | Not applicable |
| Instrument.Locale.EndMonth | Daylight savings: End month<br>0 = Febuary 1= February 2 = March<br>3 = April 4 = May 5 = June<br>6 = July 7 = August 8 = September<br>9 = October 10 = November 11 = December | uint8 | 10bb | 4283 | Not applicable |
| Instrument.Locale.EndOn | Week for changing to/from DST<br>0 = First 1= Second 2 = Third<br>3 = Fourth 4 = Last 5 = Second to last | uint8 | 10b9 | 4281 | Not applicable |
| Instrument.Locale.EndTime | DST end time in hours, minutes, seconds and milliseconds | time_t | 10b8 | 4280 | Set by Network.Modbus.TimeFormat |
| Instrument.Locale.Language | Language (0 = English) | uint8 | 10b0 | 4272 | Not applicable |
| Instrument.Locale.StartDay | DST start day. As Instrument.Locale.EndDay, above | uint8 | 10b6 | 4278 | Not applicable |
| Instrument.Locale.StartMonth | DST start month As Instrument.Locale.EndMonth, above | uint8 | 10b7 | 4279 | Not applicable |
| Instrument.Locale.StartOn | Start DST on. As Instrument.Locale.EndOn, above | uint8 | 10b5 | 4277 | Not applicable |
| Instrument.Locale.StartTime | DST start time. As Instrument.Locale.EndTime above | time_t | 10b4 | 4276 | Set by Network.Modbus.TimeFormat |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| Instrument.Locale.TimeZone | Time zone<br>0 = GMT - 12 hours 1 = GMT - 11 hours<br>2 = GMT - 10 hours 3 = GMT - 9 hours<br>4 = GMT - 8 hours 5 = GMT - 7 hours<br>6 = GMT - 6 hours 7 = GMT - 5 hours<br>8 = GMT - 4 hours 9 = GMT - 3.5 hours<br>10 = GMT - 3 hours 11 = GMT - 2 hours<br>12 = GMT - 1 hour 13 = GMT<br>14 = GMT + 1 hour 15 = GMT + 2 hours<br>16 = GMT + 3 hours 17 = GMT + 3.5 hours<br>18 = GMT + 4 hours 19 = GMT + 4.5 hours<br>20 = GMT + 5 hours 21 = GMT + 5.5 hours<br>22 = GMT + 5.75 hours 23 = GMT + 6 hours<br>24 = GMT + 6.5 hours 25 = GMT + 7 hours<br>26 = GMT + 8 hours 27 = GMT + 9 hours<br>28 = GMT + 9.5 hours 29 = GMT + 10 hours<br>30 = GMT + 11 hours 31 = GMT + 12 hours<br>32 = GMT + 13 hours | uint8 | 10b2 | 4274 | Not applicable |
| Instrument.Notes.Note | Operator Note | string_t | 5500 | 21760 | Not applicable |
| Instrument.Notes.Note1 | Operator note 1 | string_t | 5580 | 21888 | Not applicable |
| Instrument.Notes.Note2 | Operator note 2 | string_t | 5600 | 22016 | Not applicable |
| Instrument.Notes.Note3 | Operator note 3 | string_t | 5680 | 22144 | Not applicable |
| Instrument.Notes.Note4 | Operator note 4 | string_t | 5700 | 22272 | Not applicable |
| Instrument.Notes.Note5 | Operator note 5 | string_t | 5780 | 22400 | Not applicable |
| Instrument.Notes.Note6 | Operator note 6 | string_t | 5800 | 22528 | Not applicable |
| Instrument.Notes.Note7 | Operator note 7 | string_t | 5880 | 22656 | Not applicable |
| Instrument.Notes.Note8 | Operator note 8 | string_t | 5900 | 22784 | Not applicable |
| Instrument.Notes.Note9 | Operator note 9 | string_t | 5980 | 22912 | Not applicable |
| Instrument.Notes.Note10 | Operator note 10 | string_t | 5a00 | 23040 | Not applicable |
| Instrument.PromoteList.PromoteListName | Promote list (operator view) title | string_t | 6d07 | 27911 | Not applicable |
| Instrument.PromoteList.PromoteParam1 | Promote parameter (1) | eint32 | 10e0 | 4320 | Not applicable |
| strument.PromoteList.PromoteParam1Desc | Descriptor for promote parameter (1) | string_t | 6300 | 25344 | Not applicable |
| Instrument.PromoteList.PromoteParam2 | Promote parameter (2) | eint32 | 10e1 | 4321 | Not applicable |
| Instrument.PromoteList.PromoteParam2Desc | Descriptor for promote parameter (2) | string_t | 6315 | 25365 | Not applicable |
| Instrument.PromoteList.PromoteParam3 | Promote parameter (3) | eint32 | 10e2 | 4322 | Not applicable |
| Instrument.PromoteList.PromoteParam3Desc | Descriptor for promote parameter (3) | string_t | 632a | 25386 | Not applicable |
| Instrument.PromoteList.PromoteParam4 | Promote parameter (4) | eint32 | 10e3 | 4323 | Not applicable |
| Instrument.PromoteList.PromoteParam4Desc | Descriptor for promote parameter (4) | string_t | 633f | 25407 | Not applicable |
| Instrument.PromoteList.PromoteParam5 | Promote parameter (5) | eint32 | 10e4 | 4324 | Not applicable |
| Instrument.PromoteList.PromoteParam5Desc | Descriptor for promote parameter (5) | string_t | 6354 | 25428 | Not applicable |
| Instrument.PromoteList.PromoteParam6 | Promote parameter (6) | eint32 | 10e5 | 4325 | Not applicable |
| Instrument.PromoteList.PromoteParam6Desc | Descriptor for promote parameter (6) | string_t | 6369 | 25449 | Not applicable |
| Instrument.PromoteList.PromoteParam7 | Promote parameter (7) | eint32 | 10e6 | 4326 | Not applicable |
| Instrument.PromoteList.PromoteParam7Desc | Descriptor for promote parameter (7) | string_t | 637e | 25470 | Not applicable |
| Instrument.PromoteList.PromoteParam8 | Promote parameter (8) | eint32 | 10e7 | 4327 | Not applicable |
| Instrument.PromoteList.PromoteParam8Desc | Descriptor for promote parameter (8) | string_t | 6393 | 25491 | Not applicable |
| Instrument.PromoteList.PromoteParam9 | Promote parameter (9) | eint32 | 10e8 | 4328 | Not applicable |
| Instrument.PromoteList.PromoteParam9Desc | Descriptor for promote parameter (9) | string_t | 63a8 | 25512 | Not applicable |
| Instrument.PromoteList.PromoteParam10 | Promote parameter (10) | eint32 | 10e9 | 4329 | Not applicable |
| Instrument.PromoteList.PromoteParam10Desc | Descriptor for promote parameter (10) | string_t | 63bd | 25533 | Not applicable |
| InInstrument.Security.CommsPass | 1 = Password required for comms access | bool | 10c1 | 4289 | Not applicable |
| Instrument.Security.DefaultConfig | 1 = set all parameters to factory settings | bool | 10c2 | 4290 | Not applicable |
| Instrument.Security.EngineerAccess | 1 = Engineer access required | bool | 10c0 | 4288 | Not applicable |
| Instrument.Security.EngineerPassword | Engineer pass phrase (default 100) | string_t | 63d3 | 25555 | Not applicable |
| Instrument.Security.Feature2Pass | Features2 pass code | int32 | 10c4 | 4292 | Not applicable |
| Instrument.Security.Feature3Pass | Features3 pass code | int32 | 10c5 | 4293 | Not applicable |
| Instrument.Security.FeaturePass | Features pass code | int32 | 10c3 | 4291 | Not applicable |
| Instrument.Security.OEMEntry | OEM pass phrase entry | string_t | 6d61 | 28001 | Not applicable |
| Instrument.Security.OEMPass | OEM pass phrase | string_t | 6d30 | 27952 | Not applicable |
| Instrument.Security.OEMStatus | OEM status (0 = Unlocked; 1 = Locked) | bool | 10c6 | 4294 | Not applicable |
| Instrument.Security.OperatorPassword | Operator pass phrase (default = blank | string_t | 6437 | 25655 | Not applicable |
| Instrument.Security.PassPhrase | The parameter to be written to if comms security is enabled | string_t | 4416 | 17430 | Not applicable |
| Instrument.Security.SupervisorPassword | Supervisor pass phrase (default = blank | string_t | 6405 | 25605 | Not applicable |
| Lgc2.1.FallbackType | Fallback Condition<br>0 = Output False; Status Bad.<br>1 = Output True; Status Bad<br>2 = Output False; Status Good.<br>3 = Output True; Status good | uint8 | 2efb | 12027 | Not applicable |
| Lgc2.1.In1 | Input Value 1 | float32 | 2ef9 | 12025 | 0dp |
| Lgc2.1.In2 | Input Value 2 | float32 | 2efa | 12026 | 0dp |
| Lgc2.1.Invert | Sense of Input Values<br>0 = Neither input inverted<br>1 = Input 1 inverted<br>2 = Input 2 inverted<br>3 = Both inputs inverted | uint8 | 2efc | 12028 | Not applicable |
| Lgc2.1.Oper | Logic Operation<br>0 = Off; 1 = AND; 2 = OR; 3 = XOR; 4 = 1 set/2 reset<br>5 = Input 1 = Input 2? 6 = Input 1 ≠ Input 2<br>7 = Input 1 > Input 2? 8 = Input 1< Input 2?<br>9 = Input 1 ≥ Input 2?; 10 = Input 1 ≤ Input 2? | uint8 | 2ef8 | 12024 | Not applicable |
| Lgc2.1.Out | Output Value (0 = Off (false); 1 = On (true)) | bool | 2efd | 12029 | Not applicable |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| Lgc2.1.OutputStatus | Output Status (0 = Good; 1 = Bad) | uint8 | 2efe | 12030 | Not applicable |
| Lgc2.2.FallbackType | Fallback Condition (as Lgc2.1.FallbackType) | uint8 | 2f02 | 12034 | Not applicable |
| Lgc2.2.In1 | Input Value 1 | float32 | 2f00 | 12032 | 0dp |
| Lgc2.2.In2 | Input Value 2 | float32 | 2f01 | 12033 | 0dp |
| Lgc2.2.Invert | Sense of Input Value (as Lgc2.1.Invert) | uint8 | 2f03 | 12035 | Not applicable |
| Lgc2.2.Oper | Logic Operation (as Lgc2.1.Oper) | uint8 | 2eff | 12031 | Not applicable |
| Lgc2.2.Out | The result of the logic operation (as Lgc2.1.Out) | bool | 2f04 | 12036 | Not applicable |
| Lgc2.2.OutputStatus | Output Status (0 = Good; 1 = Bad) | uint8 | 2f05 | 12037 | Not applicable |
| Lgc2.3.FallbackType | Fallback Condition (as Lgc2.1.FallbackType) | uint8 | 2f09 | 12041 | Not applicable |
| Lgc2.3.In1 | Input Value 1 | float32 | 2f07 | 12039 | 0dp |
| Lgc2.3.In2 | Input Value 2 | float32 | 2f08 | 12040 | 0dp |
| Lgc2.3.Invert | Sense of Input Value (as Lgc2.1.Invert) | uint8 | 2f0a | 12042 | Not applicable |
| Lgc2.3.Oper | Logic Operation (as Lgc2.1.Oper) | uint8 | 2f06 | 12038 | Not applicable |
| Lgc2.3.Out | The result of the logic operation (as Lgc2.1.Out) | bool | 2f0b | 12043 | Not applicable |
| Lgc2.3.OutputStatus | Output Status (0 = Good; 1 = Bad) | uint8 | 2f0c | 12044 | Not applicable |
| Lgc2.4.FallbackType | Fallback Condition (as Lgc2.1.FallbackType) | uint8 | 2f10 | 12048 | Not applicable |
| Lgc2.4.In1 | Input Value 1 | float32 | 2f0e | 12046 | 0dp |
| Lgc2.4.In2 | Input Value 2 | float32 | 2f0f | 12047 | 0dp |
| Lgc2.4.Invert | Sense of Input Value (as Lgc2.1.Invert) | uint8 | 2f11 | 12049 | Not applicable |
| Lgc2.4.Oper | Logic Operation (as Lgc2.1.Oper) | uint8 | 2f0d | 12045 | Not applicable |
| Lgc2.4.Out | The result of the logic operation (as Lgc2.1.Out) | bool | 2f12 | 12050 | Not applicable |
| Lgc2.4.OutputStatus | Output Status (0 = Good; 1 = Bad) | uint8 | 2f13 | 12051 | Not applicable |
| Lgc2.5.FallbackType | Fallback Condition (as Lgc2.1.FallbackType) | uint8 | 2f17 | 12055 | Not applicable |
| Lgc2.5.In1 | Input Value 1 | float32 | 2f15 | 12053 | 0dp |
| Lgc2.5.In2 | Input Value 2 | float32 | 2f16 | 12054 | 0dp |
| Lgc2.5.Invert | Sense of Input Value (as Lgc2.1.Invert) | uint8 | 2f18 | 12056 | Not applicable |
| Lgc2.5.Oper | Logic Operation (as Lgc2.1.Oper) | uint8 | 2f14 | 12052 | Not applicable |
| Lgc2.5.Out | The result of the logic operation (as Lgc2.1.Out) | bool | 2f19 | 12057 | Not applicable |
| Lgc2.5.OutputStatus | Output Status (0 = Good; 1 = Bad) | uint8 | 2f1a | 12058 | Not applicable |
| Lgc2.6.FallbackType | Fallback Condition (as Lgc2.1.FallbackType) | uint8 | 2f1e | 12062 | Not applicable |
| Lgc2.6.In1 | Input Value 1 | float32 | 2f1c | 12060 | 0dp |
| Lgc2.6.In2 | Input Value 2 | float32 | 2f1d | 12061 | 0dp |
| Lgc2.6.Invert | Sense of Input Value (as Lgc2.1.Invert) | uint8 | 2f1f | 12063 | Not applicable |
| Lgc2.6.Oper | Logic Operation (as Lgc2.1.Oper) | uint8 | 2f1b | 12059 | Not applicable |
| Lgc2.6.Out | The result of the logic operation (as Lgc2.1.Out) | bool | 2f20 | 12064 | Not applicable |
| Lgc2.6.OutputStatus | Output Status (0 = Good; 1 = Bad) | uint8 | 2f21 | 12065 | Not applicable |
| Lgc2.7.FallbackType | Fallback Condition (as Lgc2.1.FallbackType) | uint8 | 2f25 | 12069 | Not applicable |
| Lgc2.7.In1 | Input Value 1 | float32 | 2f23 | 12067 | 0dp |
| Lgc2.7.In2 | Input Value 2 | float32 | 2f24 | 12068 | 0dp |
| Lgc2.7.Invert | Sense of Input Value (as Lgc2.1.Invert) | uint8 | 2f26 | 12070 | Not applicable |
| Lgc2.7.Oper | Logic Operation (as Lgc2.1.Oper) | uint8 | 2f22 | 12066 | Not applicable |
| Lgc2.7.Out | The result of the logic operation (as Lgc2.1.Out) | bool | 2f27 | 12071 | Not applicable |
| Lgc2.7.OutputStatus | Output Status (0 = Good; 1 = Bad) | uint8 | 2f28 | 12072 | Not applicable |
| Lgc2.8.FallbackType | Fallback Condition (as Lgc2.1.FallbackType) | uint8 | 2f2c | 12076 | Not applicable |
| Lgc2.8.In1 | Input Value 1 | float32 | 2f2a | 12074 | 0dp |
| Lgc2.8.In2 | Input Value 2 | float32 | 2f2b | 12075 | 0dp |
| Lgc2.8.Invert | Sense of Input Value (as Lgc2.1.Invert) | uint8 | 2f2d | 12077 | Not applicable |
| Lgc2.8.Oper | Logic Operation (as Lgc2.1.Oper) | uint8 | 2f29 | 12073 | Not applicable |
| Lgc2.8.Out | The result of the logic operation (as Lgc2.1.Out) | bool | 2f2e | 12078 | Not applicable |
| Lgc2.8.OutputStatus | Output Status (0 = Good; 1 = Bad) | uint8 | 2f2f | 12079 | Not applicable |
| Lgc2.9.FallbackType | Fallback Condition (as Lgc2.1.FallbackType) | uint8 | 2f33 | 12083 | Not applicable |
| Lgc2.9.In1 | Input Value 1 | float32 | 2f31 | 12081 | 0dp |
| Lgc2.9.In2 | Input Value 2 | float32 | 2f32 | 12082 | 0dp |
| Lgc2.9.Invert | Sense of Input Value (as Lgc2.1.Invert) | uint8 | 2f34 | 12084 | Not applicable |
| Lgc2.9.Oper | Logic Operation (as Lgc2.1.Oper) | uint8 | 2f30 | 12080 | Not applicable |
| Lgc2.9.Out | The result of the logic operation (as Lgc2.1.Out) | bool | 2f35 | 12085 | Not applicable |
| Lgc2.9.OutputStatus | Output Status (0 = Good; 1 = Bad) | uint8 | 2f36 | 12086 | Not applicable |
| Lgc2.10.FallbackType | Fallback Condition (as Lgc2.1.FallbackType) | uint8 | 2f3a | 12090 | Not applicable |
| Lgc2.10.In1 | Input Value 1 | float32 | 2f38 | 12088 | 0dp |
| Lgc2.10.In2 | Input Value 2 | float32 | 2f39 | 12089 | 0dp |
| Lgc2.10.Invert | Sense of Input Value (as Lgc2.1.Invert) | uint8 | 2f3b | 12091 | Not applicable |
| Lgc2.10.Oper | Logic Operation (as Lgc2.1.Oper) | uint8 | 2f37 | 12087 | Not applicable |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| Lgc2.10.Out | The result of the logic operation (as Lgc2.1.Out) | bool | 2f3c | 12092 | Not applicable |
| Lgc2.10.OutputStatus | Output Status (0 = Good; 1 = Bad) | uint8 | 2f3d | 12093 | Not applicable |
| Lgc2.11.FallbackType | Fallback Condition (as Lgc2.1.FallbackType) | uint8 | 2f41 | 12097 | Not applicable |
| Lgc2.11.In1 | Input Value 1 | float32 | 2f3f | 12095 | 0dp |
| Lgc2.11.In2 | Input Value 2 | float32 | 2f40 | 12096 | 0dp |
| Lgc2.11.Invert | Sense of Input Value (as Lgc2.1.Invert) | uint8 | 2f42 | 12098 | Not applicable |
| Lgc2.11.Oper | Logic Operation (as Lgc2.1.Oper) | uint8 | 2f3e | 12094 | Not applicable |
| Lgc2.11.Out | The result of the logic operation (as Lgc2.1.Out) | bool | 2f43 | 12099 | Not applicable |
| Lgc2.11.OutputStatus | Output Status (0 = Good; 1 = Bad) | uint8 | 2f44 | 12100 | Not applicable |
| Lgc2.12.FallbackType | Fallback Condition (as Lgc2.1.FallbackType) | uint8 | 2f48 | 12104 | Not applicable |
| Lgc2.12.In1 | Input Value 1 | float32 | 2f46 | 12102 | 0dp |
| Lgc2.12.In2 | Input Value 2 | float32 | 2f47 | 12103 | 0dp |
| Lgc2.12.Invert | Sense of Input Value (as Lgc2.1.Invert) | uint8 | 2f49 | 12105 | Not applicable |
| Lgc2.12.Oper | Logic Operation (as Lgc2.1.Oper) | uint8 | 2f45 | 12101 | Not applicable |
| Lgc2.12.Out | The result of the logic operation (as Lgc2.1.Out) | bool | 2f4a | 12106 | Not applicable |
| Lgc2.12.OutputStatus | Output Status (0 = Good; 1 = Bad) | uint8 | 2f4b | 12107 | Not applicable |
| Lgc8.1.In1 | Input 1 Value (0 = Off; 1 = On) | bool | 2f4f | 12111 | Not applicable |
| Lgc8.1.In2 | Input 2 Value (0 = Off; 1 = On) | bool | 2f50 | 12112 | Not applicable |
| Lgc8.1.In3 | Input 3 Value (0 = Off; 1 = On) | bool | 2f51 | 12113 | Not applicable |
| Lgc8.1.In4 | Input 4 Value (0 = Off; 1 = On) | bool | 2f52 | 12114 | Not applicable |
| Lgc8.1.In5 | Input 5 Value (0 = Off; 1 = On) | bool | 2f53 | 12115 | Not applicable |
| Lgc8.1.In6 | Input 6 Value (0 = Off; 1 = On) | bool | 2f54 | 12116 | Not applicable |
| Lgc8.1.In7 | Input 7 Value (0 = Off; 1 = On) | bool | 2f55 | 12117 | Not applicable |
| Lgc8.1.In8 | Input 8 Value (0 = Off; 1 = On) | bool | 2f56 | 12118 | Not applicable |
| Lgc8.1.InInvert | Invert Selected Inputs (See also section 4.20.3)<br>Hex0001 = Invert input 1 Hex0010 = invert input 5<br>Hex0002 = Invert input 2 Hex0020 = invert input 6<br>Hex0003 = Invert input 3 Hex0030 = invert input 7<br>Hex0004 = invert input 4 Hex0040 = invert input 8 | uint8 | 2f4d | 12109 | Not applicable |
| Lgc8.1.NumIn | Number of Inputs | uint8 | 2f4e | 12110 | Not applicable |
| Lgc8.1.Oper | Logic Operation (0 = Off; 1 = AND; 2 = OR; 3 = XOR) | uint8 | 2f4c | 12108 | Not applicable |
| Lgc8.1.Out | Output Value (0 = Off (false); 1 = On (true)) | bool | 2f57 | 12119 | Not applicable |
| Lgc8.1.OutInvert | Invert the Output (0 = No; 1 = Yes) | bool | 2f58 | 12120 | Not applicable |
| Lgc8.2.In1 | Input 1 Value (0 = Off; 1 = On) | bool | 2f5c | 12124 | Not applicable |
| Lgc8.2.In2 | Input 2 Value (0 = Off; 1 = On) | bool | 2f5d | 12125 | Not applicable |
| Lgc8.2.In3 | Input 3 Value (0 = Off; 1 = On) | bool | 2f5e | 12126 | Not applicable |
| Lgc8.2.In4 | Input 4 Value (0 = Off; 1 = On) | bool | 2f5f | 12127 | Not applicable |
| Lgc8.2.In5 | Input 5 Value (0 = Off; 1 = On) | bool | 2f60 | 12128 | Not applicable |
| Lgc8.2.In6 | Input 6 Value (0 = Off; 1 = On) | bool | 2f61 | 12129 | Not applicable |
| Lgc8.2.In7 | Input 7 Value (0 = Off; 1 = On) | bool | 2f62 | 12130 | Not applicable |
| Lgc8.2.In8 | Input 8 Value (0 = Off; 1 = On) | bool | 2f63 | 12131 | Not applicable |
| Lgc8.2.InInvert | Invert Selected Inputs (as Lgc8.1.InInvert) | uint8 | 2f5a | 12122 | Not applicable |
| Lgc8.2.NumIn | Number of Inputs | uint8 | 2f5b | 12123 | Not applicable |
| Lgc8.2.Oper | Logic Operation (0 = Off; 1 = AND; 2 = OR; 3 = XOR) | uint8 | 2f59 | 12121 | Not applicable |
| Lgc8.2.Out | Output Value (as Lgc8.1.Out) | bool | 2f64 | 12132 | Not applicable |
| Lgc8.2.OutInvert | Invert the Output (0 = No; 1 = Yes) | bool | 2f65 | 12133 | Not applicable |
| Loop.1.Diag.DerivativeOutContrib | Derivative Output Contribution | float32 | 0212 | 530 | 0dp |
| Loop.1.Diag.Error | Calculated error | float32 | 020d | 525 | Same as Loop.1.Main.PV |
| Loop.1.Diag.IntegralOutContrib | Integral Output Contribution | float32 | 0211 | 529 | 0dp |
| Loop.1.Diag.LoopBreakAlarm | Loop Break (0 = No break; 1 = Break) | bool | 020f | 527 | Not applicable |
| Loop.1.Diag.LoopMode | Mode of the Loop (0 = Auto; 1 = Man; 2 = Off) | uint8 | 1691 | 5777 | Not applicable |
| Loop.1.Diag.PropOutContrib | Proportional Output Contribution | float32 | 0210 | 528 | 0dp |
| Loop.1.Diag.SBrk | Sensor Break Status ( 0 = No break; 1 = Break) | bool | 0213 | 531 | Not applicable |
| Loop.1.Diag.SchedCBH | The Scheduled Cutback High (0 = Auto) | float32 | 1695 | 5781 | 0dp |
| Loop.1.Diag.SchedCBL | The Scheduled Cutback Low (0 = Auto) | float32 | 1696 | 5782 | 0dp |
| Loop.1.Diag.SchedLPBrk | The Scheduled Loop Break Time (0 = Off) | float32 | 1698 | 5784 | 0dp |
| Loop.1.Diag.SchedMR | The Scheduled Manual Reset | float32 | 1697 | 5783 | 1dp |
| Loop.1.Diag.SchedOPHi | The Scheduled Output High Limit | float32 | 169a | 5786 | 1dp |
| Loop.1.Diag.SchedOPLo | The Scheduled Output Low Limit | float32 | 169b | 5787 | 1dp |
| Loop.1.Diag.SchedPB | The Scheduled Proportional Band | float32 | 1692 | 5778 | 1dp |
| Loop.1.Diag.SchedR2G | The Scheduled Relative Cool Gain | float32 | 1699 | 5785 | 1dp |
| Loop.1.Diag.SchedTd | The Scheduled Derivative Time (0 = Off) | float32 | 1694 | 5780 | 0dp |
| Loop.1.Diag.SchedTi | The Scheduled Integral Time (0 = Off) | float32 | 1693 | 5779 | 0dp |
| Loop.1.Diag.TargetOutVal | Target Output value | float32 | 020e | 526 | Same as Loop.1.OP.OutputHighLimit |
| Loop.1.Diag.WrkOPHi | Working Output High Limit | float32 | 0215 | 533 | 0dp |
| Loop.1.Diag.WrkOPLo | Working Output Low Limit | float32 | 0214 | 532 | 0dp |
| Loop.1.Main.ActiveOut | Working Output | float32 | 0204 | 516 | Same as Loop.1.OP.OutputHighLimit |
| Loop.1.Main.AutoMan | Auto/Manual Mode (0 = Auto; 1 = Man) | bool | 0201 | 513 | Not applicable |
| Loop.1.Main.Inhibit | Control Inhibit (0 = No; 1 = Yes) | bool | 0205 | 517 | Not applicable |
| Loop.1.Main.IntHold | Integral action inhibit. 0 = No; 1 = Yes | uint8 | 0206 | 518 | Not applicable |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| Loop1.Maim.PV | Process varaible | float32 | 0200 | 512 | 1dp |
| Loop.1.Main.TargetSP | Target Setpoint | float32 | 0202 | 514 | Same as Loop.1.Main.PV |
| Loop.1.Main.WorkingSP | Working Setpoint | float32 | 0203 | 515 | Same as Loop.1.Main.PV |
| Loop.1.OP.Ch1OnOffHysteresis | Ch1 On/Off Hysteresis in Engineering Units | float32 | 1672 | 5746 | Same as Loop.1.Main.PV |
| Loop.1.OP.Ch1Out | Channel 1 Output Value | float32 | 020b | 523 | Same as Loop.1.OP.OutputHighLimit |
| Loop.1.OP.Ch1PotBreak | Ch1 Potentiometer Break (0 = Off; 1 = On) | uint8 | 1679 | 5753 | Not applicable |
| Loop.1.OP.Ch1PotPosition | Ch1 Valve Position | float32 | 1678 | 5752 | 0dp |
| Loop.1.OP.Ch1TravelTime | Channel 1 Travel Time | float32 | 1674 | 5748 | 1dp |
| Loop.1.OP.Ch2Deadband | Channel 2 Deadband | float32 | 166f | 5743 | Same as Loop.1.OP.OutputHighLimit |
| Loop.1.OP.Ch2OnOffHysteresis | Ch2 On/Off Hysteresis in Eng Units | float32 | 1673 | 5747 | Same as Loop.1.Main.PV |
| Loop.1.OP.Ch2Out | Channel 2 (Cool) Output Value | float32 | 020c | 524 | Same as Loop.1.OP.OutputHighLimit |
| Loop.1.OP.Ch2PotBreak | Ch2 Potentiometer Break (0 = Off; 1 = On) | uint8 | 167b | 5755 | Not applicable |
| Loop.1.OP.Ch2PotPosition | Ch2 Valve Position | float32 | 167a | 5754 | 0dp |
| Loop.1.OP.Ch2TravelTime | Channel 2 Travel Time | float32 | 1675 | 5749 | 1dp |
| Loop.1.OP.CoolType | Cooling Algorithm Type<br>0 = Linear 1 = Oil 2 = Water 3 = Fan | uint8 | 1683 | 5763 | Not applicable |
| Loop.1.OP.EnablePowerFeedforward | 0 = Power Feedforward disabled; 1 = PFF enabled | uint8 | 1681 | 5761 | Not applicable |
| Loop.1.OP.FeedForwardGain | Feedforward Gain | float32 | 1685 | 5765 | 3dp |
| Loop.1.OP.FeedForwardOffset | Feedforward Offset | float32 | 1686 | 5766 | 0dp |
| Loop.1.OP.FeedForwardTrimLimit | Feedforward Trim Limit | float32 | 1687 | 5767 | 0dp |
| Loop.1.OP.FeedForwardType | Feedforward Type (0 = None; 1 = Remote; 2 = SP; 3 = PV) | uint8 | 1684 | 5764 | Not applicable |
| Loop.1.OP.FeedForwardVal | Feedforward Value | float32 | 1688 | 5768 | 0dp |
| Loop.1.OP.FF_Rem | Remote Feed Forward Input | float32 | 168d | 5773 | 0dp |
| Loop.1.OP.ForcedOP | Forced manual output value | float32 | 168f | 5775 | 1dp |
| Loop.1.OP.ManStartup | Manual Startup Mode (0 = Off; 1 = On) | bool | 1690 | 5776 | Not applicable |
| Loop.1.OP.ManualMode | Manual Output Mode (0 = Track; 1 = Step; 2 = Last MOP) | uint8 | 167f | 5759 | Not applicable |
| Loop.1.OP.ManualOutVal | Manual Output Value | float32 | 1680 | 5760 | Same as Loop.1.OP.OutputHighLimit |
| Loop.1.OP.MeasuredPower | Measured Mains Voltage | float32 | 1682 | 5762 | 0dp |
| Loop.1.OP.NudgeLower | Valve Nudge Lower (1 = Lower) | uint8 | 1677 | 5751 | Not applicable |
| Loop.1.OP.NudgeRaise | Valve Nudge Raise (1 = Raise) | uint8 | 1676 | 5750 | Not applicable |
| Loop.1.OP.OutputHighLimit | Output High Limit | float32 | 166d | 5741 | 1dp |
| Loop.1.OP.OutputLowLimit | Output Low Limit | float32 | 166e | 5742 | Same as Loop.1.OP.OutputHighLimit |
| Loop.1.OP.PotBreakMode | Potentiometer Break Mode<br>(0 = Raise; 1 = Lower; 2 = Rest: 3 = Model) | uint8 | 167c | 5756 | Not applicable |
| Loop.1.OP.Rate | Output Rate Limit Value (0 = Off) | float32 | 1670 | 5744 | 1dp |
| Loop.1.OP.RateDisable | Output Rate Limit Disable (1 = Disabled) | bool | 1671 | 5745 | Not applicable |
| Loop.1.OP.RemOPH | Remote Output High Limit | float32 | 168c | 5772 | Same as Loop.1.Main.ActiveOut |
| Loop.1.OP.RemOPL | Remote Output Low Limit | float32 | 168b | 5771 | Same as Loop.1.Main.ActiveOut |
| Loop.1.OP.SafeOutVal | Safe Output Value | float32 | 167e | 5758 | Same as Loop.1.OP.OutputHighLimit |
| Loop.1.OP.SbrkOP | The output power in sensor break | float32 | 168e | 5774 | Same as Loop.1.OP.OutputHighLimit |
| Loop.1.OP.SensorBreakMode | Sensor Break Mode (0 = SbrkOP; 1 = Hold) | uint8 | 167d | 5757 | Not applicable |
| Loop.1.OP.TrackEnable | Enable Output Tracking (0 = Disabled; 1 = Enabled) | uint8 | 168a | 5770 | Not applicable |
| Loop.1.OP.TrackOutVal | Output Track Value | float32 | 1689 | 5769 | 0dp |
| Loop.1.PID.ActiveSet | Current PID Set | uint8 | 1638 | 5688 | Not applicable |
| Loop.1.PID.Boundary1-2 | Threshold for swapping between set 1 and set 2 | float32 | 1639 | 5689 | 0dp |
| Loop.1.PID.Boundary2-3 | Threshold for swapping between set 2 and set 3 | float32 | 163a | 5690 | 0dp |
| Loop.1.PID.CutbackHigh | Cutback high value for PID set 1 (0 = Auto) | float32 | 163f | 5695 | 1dp |
| Loop.1.PID.CutbackHigh2 | Cutback high value for PID set 2 (0 = Auto) | float32 | 1647 | 5703 | 1dp |
| Loop.1.PID.CutbackHigh3 | Cutback high value for PID set 3 (0 = Auto) | float32 | 164f | 5711 | 1dp |
| Loop.1.PID.CutbackLow | Cutback low value for PID set 1 (0 = Auto) | float32 | 1640 | 5696 | 1dp |
| Loop.1.PID.CutbackLow2 | Cutback low value for PID set 2 (0 = Auto) | float32 | 1648 | 5704 | 1dp |
| Loop.1.PID.CutbackLow3 | Cutback low value for PID set 3 (0 = Auto) | float32 | 1650 | 5712 | 1dp |
| Loop.1.PID.DerivativeTime | Derivative time for PID set 1 | float32 | 163d | 5693 | 0dp |
| Loop.1.PID.DerivativeTime2 | Derivative time for PID set 2 | float32 | 1645 | 5701 | 0dp |
| Loop.1.PID.DerivativeTime3 | Derivative time for PID set 3 | float32 | 164d | 5709 | 0dp |
| Loop.1.PID.IntegralTime | Integral time for PID set 1 | float32 | 163c | 5692 | 0dp |
| Loop.1.PID.IntegralTime2 | Integral time for PID set 2 | float32 | 1644 | 5700 | 0dp |
| Loop.1.PID.IntegralTime3 | Integral time for PID set 3 | float32 | 164c | 5708 | 0dp |
| Loop.1.PID.LoopBreakTime | Loop break time for PID set 1 | float32 | 1642 | 5698 | 0dp |
| Loop.1.PID.LoopBreakTime2 | Loop break time for PID set 2 | float32 | 164a | 5706 | 0dp |
| Loop.1.PID.LoopBreakTime3 | Loop break time for PID set 3 | float32 | 1652 | 5714 | 0dp |
| Loop.1.PID.ManualReset | Manual reset value for PID set 1 | float32 | 1641 | 5697 | 1dp |
| Loop.1.PID.ManualReset2 | Manual reset value for PID set 2 | float32 | 1649 | 5705 | 1dp |
| Loop.1.PID.ManualReset3 | Manual reset value for PID set 3 | float32 | 1651 | 5713 | 1dp |
| Loop.1.PID.NumSets | Number of PID Sets to be used (max = 3) | uint8 | 1636 | 5686 | Not applicable |
| Loop.1.PID.OutputHi | Gain scheduled output high limit for PID set 1 | float32 | 1653 | 5715 | 1dp |
| Loop.1.PID.OutputHi2 | Gain scheduled output high limit for PID set 2 | float32 | 1655 | 5717 | 1dp |
| Loop.1.PID.OutputHi3 | Gain scheduled output high limit for PID set 3 | float32 | 1657 | 5719 | 1dp |
| Loop.1.PID.OutputLo | Gain scheduled output low limit for PID set 1 | float32 | 1654 | 5716 | 1dp |
| Loop.1.PID.OutputLo2 | Gain scheduled output low limit for PID set 2 | float32 | 1656 | 5718 | 1dp |
| Loop.1.PID.OutputLo3 | Gain scheduled output low limit for PID set 3 | float32 | 1658 | 5720 | 1dp |
| Loop.1.PID.ProportionalBand | Proportional band value for PID set 1 | float32 | 163b | 5691 | 1dp |
| Loop.1.PID.ProportionalBand2 | Proportional band value for PID set 2 | float32 | 1643 | 5699 | 1dp |
| Loop.1.PID.ProportionalBand3 | Proportional band value for PID set 3 | float32 | 164b | 5707 | 1dp |
| Loop.1.PID.RelCh2Gain | Channel 2 relative cool gain value for PID set 1 | float32 | 163e | 5694 | 1dp |
| Loop.1.PID.RelCh2Gain2 | Channel 2 relative cool gain value for PID set 2 | float32 | 1646 | 5702 | 1dp |
| Loop.1.PID.RelCh2Gain3 | Channel 2 relative cool gain value for PID set 3 | float32 | 164e | 5710 | 1dp |
| Loop.1.PID.SchedulerRemoteInput | Scheduler Remote Input | float32 | 1637 | 5687 | 0dp |
| Loop.1.PID.SchedulerType | Scheduler Type<br>0 = Off 1 = Set 2 = SP 3 = PV<br>4 = Error 5 = OP 6 = Rem | uint8 | 1635 | 5685 | Not applicable |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| Loop.1.Setup.AutoManAccess | Edit access to 'Auto Man' in Loop display page<br>0 = Read/Write (R/W) all modes<br>1 = Editable in all modes except 'Logged out'<br>2 = Editable only at Engineer and Supervisor levels | uint8 | 16a8 | 5800 | Not applicable |
| Loop.1.Setup.CH1ControlType | Heat/Ch1 Control Type<br>0 =Off; 1 = On Off; 2 = PID; 3 = VPU; 4 = VPB | uint8 | 1601 | 5633 | Not applicable |
| Loop.1.Setup.CH2ControlType | Channel 2 control type (As channel 1, above) | uint8 | 1602 | 5634 | Not applicable |
| Loop.1.Setup.ControlAction | Control Action (0 = Reverse; 1 = Direct) | uint8 | 1603 | 5635 | Not applicable |
| Loop.1.Setup.DerivativeType | Derivative Type (0 = PV; 1 = Error) | uint8 | 1605 | 5637 | Not applicable |
| Loop.1.Setup.LoopName | Loop Name | string_t | 5d00 | 23808 | Not applicable |
| Loop.1.Setup.LoopType | Loop Type (0 = Single; 1 = Cascade; 2 = Override; 3 = Ratio) | uint8 | 1600 | 5632 | Not applicable |
| Loop.1.Setup.PBUnits | Proportional Band Units | uint8 | 1604 | 5636 | Not applicable |
| Loop.1.Setup.SPAccess | Edit access to 'SP' in Loop display page<br>0 = Read/Write (R/W) all modes<br>1 = Editable in all modes except 'Logged out'<br>2 = Editable only at Engineer and Supervisor levels | uint8 | 16a7 | 5799 | Not applicable |
| Loop.1.SP.AltSP | Alternative Setpoint | float32 | 1660 | 5728 | Same as Loop.1.Main.PV |
| Loop.1.SP.AltSPSelect | Alternative Setpoint Enable (0 = disable; 1 = enable) | uint8 | 1661 | 5729 | Not applicable |
| Loop.1.SP.ManualTrack | Manual Track Enable (0 = disable; 1 = enable) | uint8 | 1667 | 5735 | Not applicable |
| Loop.1.SP.RangeHigh | Setpoint Range High Limit | float32 | 1659 | 5721 | Same as Loop.1.Main.PV |
| Loop.1.SP.RangeLow | Setpoimt Range Low Limit | float32 | 165a | 5722 | Same as Loop.1.Main.PV |
| Loop.1.SP.Rate | Setpoint Rate Limit Value (0 = Rate limit off) | float32 | 1662 | 5730 | Same as Loop.1.Main.PV |
| Loop.1.SP.RateDisable | Setpoint Rate Limit Disable (0 = No; 1 = Yes) | bool | 1663 | 5731 | Not applicable |
| Loop.1.SP.RateDone | Setpoint Rate Limit Complete (0 = No; 1 = Yes) | bool | 020a | 522 | Not applicable |
| Loop.1.SP.ServoToPV | Servo to PV Enable (0 = No; 1 = Yes) | bool | 166c | 5740 | Not applicable |
| Loop.1.SP.SP1 | Setpoint 1 | float32 | 165c | 5724 | Same as Loop.1.Main.PV |
| Loop.1.SP.SP2 | Setpoint 2 | float32 | 165d | 5725 | Same as Loop.1.Main.PV |
| Loop.1.SP.SPHighLimit | Setpoint High Limit | float32 | 165e | 5726 | Same as Loop.1.Main.PV |
| Loop.1.SP.SPIntBal | SP Integral Balance (0 = Off; 1 = On) | bool | 166b | 5739 | Not applicable |
| Loop.1.SP.SPLowLimit | Setpoint Low Limit | float32 | 165f | 5727 | Same as Loop.1.Main.PV |
| Loop.1.SP.SPSelect | Active Setpoint Select (0 = SP1; 1 = SP2) | uint8 | 165b | 5723 | Not applicable |
| Loop.1.SP.SPTrack | Enables setpoint tracking (0 = Off; 1 = On) | uint8 | 1668 | 5736 | Not applicable |
| Loop.1.SP.SPTrim | Setpoint Trim value | float32 | 1664 | 5732 | Same as Loop.1.Main.PV |
| Loop.1.SP.SPTrimHighLimit | Setpoint Trim High Limit | float32 | 1665 | 5733 | Same as Loop.1.Main.PV |
| Loop.1.SP.SPTrimLowLimit | Setpoint Trim Low Limit | float32 | 1666 | 5734 | Same as Loop.1.Main.PV |
| Loop.1.SP.TrackPV | Track PV | float32 | 1669 | 5737 | Same as Loop.1.Main.PV |
| Loop.1.SP.TrackSP | Manual Tracking Value | float32 | 166a | 5738 | Same as Loop.1.Main.PV |
| Loop.1.Tune.Alpha | Alpha | float32 | 16ad | 5805 | 4dp |
| Loop.1.Tune.Alpha_p | Alpha_p | float32 | 16ab | 5803 | 2dp |
| Loop.1.Tune.AutotuneEnable | Autotune Enable (0 = Autotune Off; 1 = on) | bool | 1631 | 5681 | Not applicable |
| Loop.1.Tune.CycleNo | CycleNo | float32 | 16af | 5807 | 0dp |
| Loop.1.Tune.Debug | Debug | float32 | 16ae | 5806 | 2dp |
| Loop.1.Tune.Diagnostics | Tuning diagnostics | bool | 31cd | 12749 | Not applicable |
| Loop.1.Tune.OPss | OPss | float32 | 16ac | 5804 | 2dp |
| Loop.1.Tune.OutputHighLimit | Autotune High Output Power Limit | float32 | 1632 | 5682 | Same as Loop.1.OP.OutputHighLimit |
| Loop.1.Tune.OutputLowLimit | Autotune Low Output Power Limit | float32 | 1633 | 5683 | Same as Loop.1.OP.OutputHighLimit |
| Loop.1.Tune.PBs | PBs | float32 | 16b0 | 5808 | 2dp |
| Loop.1.Tune.Settle | Settle | float32 | 16b2 | 5810 | 2dp |
| Loop.1.Tune.Stage | Autotune stage<br>0 = Reset 1 = None 2 = Monitor<br>3 = Current SP 4 = NewSP 5 = ToSp<br>6 = Max 7 = Min | uint8 | 0208 | 520 | Not applicable |
| Loop.1.Tune.StageTime | Time in this Stage of Tune | float32 | 0209 | 521 | 0dp |
| Loop.1.Tune.State | Tune status<br>0 = Off 1 = Ready 2 = Running<br>3 = Complete 4 = Timeout 5 = Ti Lmit<br>6 = R2g limit | uint8 | 0207 | 519 | Not applicable |
| Loop.1.Tune.TDs | TDs | float32 | 16b1 | 5809 | 2dp |
| Loop.1.Tune.TuneR2G | R2G Tuning Type | uint8 | 1607 | 5639 | Not applicable |
| Loop.1.Tune.Tuning | Tuning | float32 | 16aa | 5802 | 0dp |
| Loop.1.Tune.Type | Autotune Algorithm Type<br>(0 = Cycle; 1 = Single; 2 = Adaptive; 3 = R2GPD) | uint8 | 1630 | 5680 | Not applicable |
| Loop.2.Diag.DerivativeOutContrib | Derivative Output Contribution | float32 | 0292 | 658 | 0dp |
| Loop.2.Diag.Error | Calculated Error | float32 | 028d | 653 | Same as Loop.2.Main.PV |
| Loop.2.Diag.IntegralOutContrib | Integral Output Contribution | float32 | 0291 | 657 | 0dp |
| Loop.2.Diag.LoopBreakAlarm | Loop Break (0 = No break; 1 = Break) | bool | 028f | 655 | Not applicable |
| Loop.2.Diag.LoopMode | Loop mode (0 = Auto; 1 = Man; 2 = Off) | uint8 | 1791 | 6033 | Not applicable |
| Loop.2.Diag.PropOutContrib | Proportional Output Contribution | float32 | 0290 | 656 | 0dp |
| Loop.2.Diag.SBrk | Sensor break status (0 = No break; 1 = Break) | bool | 0293 | 659 | Not applicable |
| Loop.2.Diag.SchedCBH | The Scheduled Cutback Hi (0 = Auto) | float32 | 1795 | 6037 | 0dp |
| Loop.2.Diag.SchedCBL | The Scheduled Cutback Lo (0 = Auto) | float32 | 1796 | 6038 | 0dp |
| Loop.2.Diag.SchedLPBrk | The Scheduled Loop Break Time | float32 | 1798 | 6040 | 0dp |
| Loop.2.Diag.SchedMR | The Scheduled Manual Reset | float32 | 1797 | 6039 | 1dp |
| Loop.2.Diag.SchedOPHi | The Scheduled Output High Limit | float32 | 179a | 6042 | 1dp |
| Loop.2.Diag.SchedOPLo | The Scheduled Output Low Limit | float32 | 179b | 6043 | 1dp |
| Loop.2.Diag.SchedPB | The Scheduled Proportional Band | float32 | 1792 | 6034 | 1dp |
| Loop.2.Diag.SchedR2G | The Scheduled Relative Cool Gain | float32 | 1799 | 6041 | 1dp |
| Loop.2.Diag.SchedTd | The Scheduled Derivative Time (0 = Off) | float32 | 1794 | 6036 | 0dp |
| Loop.2.Diag.SchedTi | The Scheduled Integral Time (0 = Off) | float32 | 1793 | 6035 | 0dp |
| Loop.2.Diag.TargetOutVal | Target Output | float32 | 028e | 654 | Same as Loop.2.OP.OutputHighLimit |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| Loop.2.Diag.WrkOPHi | Working Output Hi Limit | float32 | 0295 | 661 | 0dp |
| Loop.2.Diag.WrkOPLo | Working Output Lo Limit | float32 | 0294 | 660 | 0dp |
| Loop.2.Main.ActiveOut | Working Output | float32 | 0284 | 644 | Same as Loop.2.OP.OutputHighLimit |
| Loop.2.Main.AutoMan | Auto/Manual Mode (Mode. 0 = Auto; 1 = Man) | bool | 0281 | 641 | Not applicable |
| Loop.2.Main.Inhibit | Control Inhibit (0 = No; 1 = Yes) | bool | 0285 | 645 | Not applicable |
| Loop.2.Main.IntHold | Integral action inhibit. 0 = No; 1 = Yes | uint8 | 0286 | 646 | Not applicable |
| Loop.2.Main.PV | Process Variable value | float32 | 0280 | 640 | 1dp |
| Loop.2.Main.TargetSP | Target Setpoint | float32 | 0282 | 642 | Same as Loop.2.Main.PV |
| Loop.2.Main.WorkingSP | Working Setpoint | float32 | 0283 | 643 | Same as Loop.2.Main.PV |
| Loop.2.OP.Ch1OnOffHysteresis | Channel 1 hysteresis in engineering units | float32 | 1772 | 6002 | Same as Loop.2.Main.PV |
| Loop.2.OP.Ch1Out | Channel 1 Output Value | float32 | 028b | 651 | Same as Loop.2.OP.OutputHighLimit |
| Loop.2.OP.Ch1PotBreak | Ch1 Potentiometer Break (0 = Off; 1 = On) | uint8 | 1779 | 6009 | Not applicable |
| Loop.2.OP.Ch1PotPosition | Ch1 Valve Position | float32 | 1778 | 6008 | 0dp |
| Loop.2.OP.Ch1TravelTime | Channel 1 Travel Time | float32 | 1774 | 6004 | 1dp |
| Loop.2.OP.Ch2Deadband | Channel 2 Deadband | float32 | 176f | 5999 | Same as Loop.2.OP.OutputHighLimit |
| Loop.2.OP.Ch2OnOffHysteresis | Channel 2 hysteresis in engineering units | float32 | 1773 | 6003 | Same as Loop.2.Main.PV |
| Loop.2.OP.Ch2Out | Channel 2 output value | float32 | 028c | 652 | Same as Loop.2.OP.OutputHighLimit |
| Loop.2.OP.Ch2PotBreak | Channel 2 Potentiometer Break (0 = Off; 1 = On) | uint8 | 177b | 6011 | Not applicable |
| Loop.2.OP.Ch2PotPosition | Channel 2 Valve Position | float32 | 177a | 6010 | 0dp |
| Loop.2.OP.Ch2TravelTime | Channel 2 Travel Time | float32 | 1775 | 6005 | 1dp |
| Loop.2.OP.CoolType | Cooling Algorithm Type<br>0 = Linear; 1 = Oil; 2 = Water; 3 = Fan | uint8 | 1783 | 6019 | Not applicable |
| Loop.2.OP.EnablePowerFeedforward | 0 = Power Feedforward disabled; 1 = PFF enabled | uint8 | 1781 | 6017 | Not applicable |
| Loop.2.OP.FeedForwardGain | Feedforward Gain | float32 | 1785 | 6021 | 3dp |
| Loop.2.OP.FeedForwardOffset | Feedforward Offset | float32 | 1786 | 6022 | 0dp |
| Loop.2.OP.FeedForwardTrimLimit | Feedforward Trim Limit | float32 | 1787 | 6023 | 0dp |
| Loop.2.OP.FeedForwardType | Feedforward Type (0 = None; 1 = Remote; 2 = SP; 3 = PV) | uint8 | 1784 | 6020 | Not applicable |
| Loop.2.OP.FeedForwardVal | Feedforward Value | float32 | 1788 | 6024 | 0dp |
| Loop.2.OP.FF_Rem | Remote Feed Forward Input | float32 | 178d | 6029 | 0dp |
| Loop.2.OP.ForcedOP | Forced manual output value | float32 | 178f | 6031 | 1dp |
| Loop.2.OP.ManStartup | Manual Startup Mode (0 = Off; 1 = On) | bool | 1790 | 6032 | Not applicable |
| Loop.2.OP.ManualMode | Manual Output Mode (0 = Track; 1 = Step; 2 = Last MOP) | uint8 | 177f | 6015 | Not applicable |
| Loop.2.OP.ManualOutVal | Manual Output Value | float32 | 1780 | 6016 | Same as Loop.2.OP.OutputHighLimit |
| Loop.2.OP.MeasuredPower | Measured Mains Voltage | float32 | 1782 | 6018 | 0dp |
| Loop.2.OP.NudgeLower | Valve Nudge Lower (1 = Lower) | uint8 | 1777 | 6007 | Not applicable |
| Loop.2.OP.NudgeRaise | Valve Nudge Raise (1 = Raise) | uint8 | 1776 | 6006 | Not applicable |
| Loop.2.OP.OutputHighLimit | Output High Limit | float32 | 176d | 5997 | 1dp |
| Loop.2.OP.OutputLowLimit | Output Low Limit | float32 | 176e | 5998 | Same as Loop.2.OP.OutputHighLimit |
| Loop.2.OP.PotBreakMode | Potentiometer Break Mode<br>(0 = Raise; 1 = Lower; 2 = Rest; 3 = Model) | uint8 | 177c | 6012 | Not applicable |
| Loop.2.OP.Rate | Output Rate Limit Value (0 = off) | float32 | 1770 | 6000 | 1dp |
| Loop.2.OP.RateDisable | Output Rate Limit Disable (0 = No, 1 = Yes) | bool | 1771 | 6001 | Not applicable |
| Loop.2.OP.RemOPH | Remote Output High Limit | float32 | 178c | 6028 | Same as Loop.2.Main.ActiveOut |
| Loop.2.OP.RemOPL | Remote Output Low Limit | float32 | 178b | 6027 | Same as Loop.2.Main.ActiveOut |
| Loop.2.OP.SafeOutVal | Safe Output Value | float32 | 177e | 6014 | Same as Loop.2.OP.OutputHighLimit |
| Loop.2.OP.SbrkOP | The output power under sensor break conditions | float32 | 178e | 6030 | Same as Loop.2.OP.OutputHighLimit |
| Loop.2.OP.SensorBreakMode | Sensor Break Mode (0 = SbrkOP; 1 = Hold) | uint8 | 177d | 6013 | Not applicable |
| Loop.2.OP.TrackEnable | Enable Output Tracking (0 = Off; 1 = On) | uint8 | 178a | 6026 | Not applicable |
| Loop.2.OP.TrackOutVal | Output Track Value | float32 | 1789 | 6025 | 0dp |
| Loop.2.PID.ActiveSet | Current PID set | uint8 | 1738 | 5944 | Not applicable |
| Loop.2.PID.Boundary1-2 | Threshold for swapping between set 1 and set 2 | float32 | 1739 | 5945 | 0dp |
| Loop.2.PID.Boundary2-3 | Threshold for swapping between set 2 and set 3 | float32 | 173a | 5946 | 0dp |
| Loop.2.PID.CutbackHigh | Cutback high value for PID set 1 (0 = Auto) | float32 | 173f | 5951 | 1dp |
| Loop.2.PID.CutbackHigh2 | Cutback high value for PID set 2 (0 = Auto) | float32 | 1747 | 5959 | 1dp |
| Loop.2.PID.CutbackHigh3 | Cutback high value for PID set 3 (0 = Auto) | float32 | 174f | 5967 | 1dp |
| Loop.2.PID.CutbackLow | Cutback low value for PID set 1 (0 = Auto) | float32 | 1740 | 5952 | 1dp |
| Loop.2.PID.CutbackLow2 | Cutback low value for PID set 2 (0 = Auto) | float32 | 1748 | 5960 | 1dp |
| Loop.2.PID.CutbackLow3 | Cutback low value for PID set 3 (0 = Auto) | float32 | 1750 | 5968 | 1dp |
| Loop.2.PID.DerivativeTime | Derivative time for PID set 1 | float32 | 173d | 5949 | 0dp |
| Loop.2.PID.DerivativeTime2 | Derivative time for PID set 2 | float32 | 1745 | 5957 | 0dp |
| Loop.2.PID.DerivativeTime3 | Derivative time for PID set 3 | float32 | 174d | 5965 | 0dp |
| Loop.2.PID.IntegralTime | Integral time for PID set 1 | float32 | 173c | 5948 | 0dp |
| Loop.2.PID.IntegralTime2 | Integral time for PID set 2 | float32 | 1744 | 5956 | 0dp |
| Loop.2.PID.IntegralTime3 | Integral time for PID set 3 | float32 | 174c | 5964 | 0dp |
| Loop.2.PID.LoopBreakTime | Loop break time for PID set 1 | float32 | 1742 | 5954 | 0dp |
| Loop.2.PID.LoopBreakTime2 | Loop break time for PID set 2 | float32 | 174a | 5962 | 0dp |
| Loop.2.PID.LoopBreakTime3 | Loop break time for PID set 3 | float32 | 1752 | 5970 | 0dp |
| Loop.2.PID.ManualReset | Manual reset value for PID set 1 | float32 | 1741 | 5953 | 1dp |
| Loop.2.PID.ManualReset2 | Manual reset value for PID set 2 | float32 | 1749 | 5961 | 1dp |
| Loop.2.PID.ManualReset3 | Manual reset value for PID set 3 | float32 | 1751 | 5969 | 1dp |
| Loop.2.PID.NumSets | Number of PID sets to be used (max. 3) | uint8 | 1736 | 5942 | Not applicable |
| Loop.2.PID.OutputHi | Gain scheduled output high limit for PID set 1 | float32 | 1753 | 5971 | 1dp |
| Loop.2.PID.OutputHi2 | Gain scheduled output high limit for PID set 2 | float32 | 1755 | 5973 | 1dp |
| Loop.2.PID.OutputHi3 | Gain scheduled output high limit for PID set 3 | float32 | 1757 | 5975 | 1dp |
| Loop.2.PID.OutputLo | Gain scheduled output low limit for PID set 1 | float32 | 1754 | 5972 | 1dp |
| Loop.2.PID.OutputLo2 | Gain scheduled output low limit for PID set 2 | float32 | 1756 | 5974 | 1dp |
| Loop.2.PID.OutputLo3 | Gain scheduled output low limit for PID set 3 | float32 | 1758 | 5976 | 1dp |
| Loop.2.PID.ProportionalBand | Proportional band value for PID set 1 | float32 | 173b | 5947 | 1dp |
| Loop.2.PID.ProportionalBand2 | Proportional band value for PID set 2 | float32 | 1743 | 5955 | 1dp |
| Loop.2.PID.ProportionalBand3 | Proportional band value for PID set 3 | float32 | 174b | 5963 | 1dp |
| Loop.2.PID.RelCh2Gain | Channel 2 relative cool gain value for PID set 1 | float32 | 173e | 5950 | 1dp |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| Loop.2.PID.RelCh2Gain2 | Channel 2 relative cool gain value for PID set 2 | float32 | 1746 | 5958 | 1dp |
| Loop.2.PID.RelCh2Gain3 | Channel 2 relative cool gain value for PID set 3 | float32 | 174e | 5966 | 1dp |
| Loop.2.PID.SchedulerRemoteInput | Scheduler Remote Input | float32 | 1737 | 5943 | 0dp |
| Loop.2.PID.SchedulerType | Scheduler Type<br>0 = Off 1 = Set 2 = SP 3 = PV<br>4 = Error 5 = OP 6 = Rem | uint8 | 1735 | 5941 | Not applicable |
| Loop.2.Setup.AutoManAccess | Edit access to 'Auto Man' in Loop display page<br>0 = Read/Write (R/W) all modes<br>1 = Editable in all modes except 'Logged out'<br>2 = Editable only at Engineer and Supervisor levels | uint8 | 17a8 | 6056 | Not applicable |
| Loop.2.Setup.CH1ControlType | Channel 1 Control Type<br>0 =Off; 1 = On Off; 2 = PID; 3 = VPU; 4 = VPB | uint8 | 1701 | 5889 | Not applicable |
| Loop.2.Setup.CH2ControlType | Channel 2 Control Type (As channel 1, above) | uint8 | 1702 | 5890 | Not applicable |
| Loop.2.Setup.ControlAction | Control Action (0 = Reverse; 1 = Direct) | uint8 | 1703 | 5891 | Not applicable |
| Loop.2.Setup.DerivativeType | Derivative Type (0 = PV; 1 = Error) | uint8 | 1705 | 5893 | Not applicable |
| Loop.2.Setup.LoopName | Loop Name | string_t | 5d10 | 23824 | Not applicable |
| Loop.2.Setup.LoopType | Loop Type (0 = single; 1 = cascade; 2 = override; 3 = ratio) | uint8 | 1700 | 5888 | Not applicable |
| Loop.2.Setup.PBUnits | Proportional Band Units (0 = Engineering units; 1 = percent) | uint8 | 1704 | 5892 | Not applicable |
| Loop.2.Setup.SPAccess | Edit access to 'SP' in Loop display page<br>0 = Read/Write (R/W) all modes<br>1 = Editable in all modes except 'Logged out'<br>2 = Editable only at Engineer and Supervisor levels | uint8 | 17a7 | 6055 | Not applicable |
| Loop.2.SP.AltSP | Alternative Setpoint | float32 | 1760 | 5984 | Same as Loop.2.Main.PV |
| Loop.2.SP.AltSPSelect | Select alternative setpoint (0 = No; 1 = Yes) | uint8 | 1761 | 5985 | Not applicable |
| Loop.2.SP.ManualTrack | Manual Track Enable (0 = disable; 1 = enable) | uint8 | 1767 | 5991 | Not applicable |
| Loop.2.SP.RangeHigh | Setpoint Range High Limit | float32 | 1759 | 5977 | Same as Loop.2.Main.PV |
| Loop.2.SP.RangeLow | Setpoint Range Low Limit | float32 | 175a | 5978 | Same as Loop.2.Main.PV |
| Loop.2.SP.Rate | Setpoint Rate Limit Value (0 = Rate limit off) | float32 | 1762 | 5986 | Same as Loop.2.Main.PV |
| Loop.2.SP.RateDisable | Setpoint Rate Limit Disable (0 = No; 1 = Yes) | bool | 1763 | 5987 | Not applicable |
| Loop.2.SP.RateDone | Setpoint Rate Limit Complete (0 = No; 1 = Yes) | bool | 028a | 650 | Not applicable |
| Loop.2.SP.ServoToPV | Servo to PV Enable (0 = No; 1 = Yes) | bool | 176c | 5996 | Not applicable |
| Loop.2.SP.SP1 | Setpoint 1 | float32 | 175c | 5980 | Same as Loop.2.Main.PV |
| Loop.2.SP.SP2 | Setpoint 2 | float32 | 175d | 5981 | Same as Loop.2.Main.PV |
| Loop.2.SP.SPHighLimit | Setpoint High Limit | float32 | 175e | 5982 | Same as Loop.2.Main.PV |
| Loop.2.SP.SPIntBal | SP Integral Balance (0 = Off; 1 = On) | bool | 176b | 5995 | Not applicable |
| Loop.2.SP.SPLowLimit | Setpoint Low Limit | float32 | 175f | 5983 | Same as Loop.2.Main.PV |
| Loop.2.SP.SPSelect | Active Setpoint Select (0 = SP1; 1 = SP2) | uint8 | 175b | 5979 | Not applicable |
| Loop.2.SP.SPTrack | Enables setpoint tracking (0 = Off; 1 = On) | uint8 | 1768 | 5992 | Not applicable |
| Loop.2.SP.SPTrim | Setpoint Trim | float32 | 1764 | 5988 | Same as Loop.2.Main.PV |
| Loop.2.SP.SPTrimHighLimit | Setpoint Trim High Limit | float32 | 1765 | 5989 | Same as Loop.2.Main.PV |
| Loop.2.SP.SPTrimLowLimit | Setpoint Trim Low Limit | float32 | 1766 | 5990 | Same as Loop.2.Main.PV |
| Loop.2.SP.TrackPV | PV for Programmer to Track | float32 | 1769 | 5993 | Same as Loop.2.Main.PV |
| Loop.2.SP.TrackSP | Manual Tracking Value | float32 | 176a | 5994 | Same as Loop.2.Main.PV |
| Loop.2.Tune.Alpha | Alpha | float32 | 17ad | 6061 | 4dp |
| Loop.2.Tune.Alpha_p | Alpha_p | float32 | 17ab | 6059 | 2dp |
| Loop.2.Tune.AutotuneEnable | Initiate autotune (0 = Autotune Off; 1 = on) | bool | 1731 | 5937 | Not applicable |
| Loop.2.Tune.CycleNo | CycleNo | float32 | 17af | 6063 | 0dp |
| Loop.2.Tune.Debug | Debug | float32 | 17ae | 6062 | 2dp |
| Loop.2.Tune.Diagnostics | Tuning diagnostics | bool | 31ce | 12750 | Not applicable |
| Loop.2.Tune.OPss | OPss | float32 | 17ac | 6060 | 2dp |
| Loop.2.Tune.OutputHighLimit | Autotune High Output Power Limit | float32 | 1732 | 5938 | Same as Loop.2.OP.OutputHighLimit |
| Loop.2.Tune.OutputLowLimit | Autotune Low Output Power Limit | float32 | 1733 | 5939 | Same as Loop.2.OP.OutputHighLimit |
| Loop.2.Tune.PBs | PBs | float32 | 17b0 | 6064 | 2dp |
| Loop.2.Tune.Settle | Settle | float32 | 17b2 | 6066 | 2dp |
| Loop.2.Tune.Stage | Stage of Tune<br>0 = Reset 1 = None 2 = Monitor<br>3 = Current SP 4 = NewSP 5 = ToSp<br>6 = Max 7 = Min | uint8 | 0288 | 648 | Not applicable |
| Loop.2.Tune.StageTime | Time in this Stage of Tune | float32 | 0289 | 649 | 0dp |
| Loop.2.Tune.State | Autotune state<br>0 = Off 1 = Ready 2 = Complete<br>3 = Timeout 4 = Ti Lmit 5 = R2g limit | uint8 | 0287 | 647 | Not applicable |
| Loop.2.Tune.TDs | TDs | float32 | 17b1 | 6065 | 2dp |
| Loop.2.Tune.TuneR2G | R2G Tuning Type | uint8 | 1608 | 5640 | Not applicable |
| Loop.2.Tune.Tuning | Tuning | float32 | 17aa | 6058 | 0dp |
| Loop.2.Tune.Type | Autotune Algorithm Type<br>(0 = Cycle; 1 = Single; 2 = Adaptive; 3 = R2GPD) | uint8 | 1730 | 5936 | Not applicable |
| Math2.1.Fallback | Fallback strategy<br>0 = Clip Bad; 1 = Clip Good; 2 = Fallback Bad<br>3 = Fallback Good; 4 = Up scale; 5 = Down scale. | uint8 | 2faf | 12207 | Not applicable |
| Math2.1.FallbackVal | Fallback Value | float32 | 2fab | 12203 | Same as Math2.1.Out |
| Math2.1.HighLimit | Output High Limit | float32 | 2fac | 12204 | Same as Math2.1.Out |
| Math2.1.In1 | Input 1 Value | float32 | 2fa7 | 12199 | 0dp |
| Math2.1.In1Mul | Input 1 Multiplier | float32 | 2fa6 | 12198 | 1dp |
| Math2.1.In2 | Input 2 Value | float32 | 2fa9 | 12201 | 0dp |
| Math2.1.In2Mul | Input 2 Multiplier | float32 | 2fa8 | 12200 | 1dp |
| Math2.1.LowLimit | Output Low Limit | float32 | 2fad | 12205 | Same as Math2.1.Out |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| Math2.1.Oper | Operation<br>0 = Off 1 = Add 2 = Subtract<br>3 = Multiply 4 =Divide 5 = Abs diff<br>6 = Select Max 7 = Select Min 8 = Hot Swap<br>9 = Sample & Hold 10 = Power 11 = Square root<br>12 = Log 13 = Ln 14 = Exponential<br>15 = 10 to the X 51 = Select | uint8 | 2faa | 12202 | Not applicable |
| Math2.1.Out | Output Value | float32 | 2fae | 12206 | Set by Math2.1.Resolution |
| Math2.1.Resolution | Output Resolution | uint8 | 2fb2 | 12210 | Not applicable |
| Math2.1.Select | Select Input 1 or Input 2 | bool | 2fb0 | 12208 | Not applicable |
| Math2.1.Status | Status. 0 = Good (OK); 7 = Bad (Error) | uint8 | 2fb1 | 12209 | Not applicable |
| Math2.1.Units | Output Units | string_t | 6944 | 26948 | Not applicable |
| | | | | | |
| Math2.2.Fallback | Fallback strategy (as Math2.1.Fallback) | uint8 | 2fbc | 12220 | Not applicable |
| Math2.2.FallbackVal | Fallback Value | float32 | 2fb8 | 12216 | Same as Math2.2.Out |
| Math2.2.HighLimit | Output High Limit | float32 | 2fb9 | 12217 | Same as Math2.2.Out |
| Math2.2.In1 | Input 1 Value | float32 | 2fb4 | 12212 | 0dp |
| Math2.2.In1Mul | Input 1 Scale | float32 | 2fb3 | 12211 | 1dp |
| Math2.2.In2 | Input 2 Value | float32 | 2fb6 | 12214 | 0dp |
| Math2.2.In2Mul | Input 2 Scale | float32 | 2fb5 | 12213 | 1dp |
| Math2.2.LowLimit | Output Low Limit | float32 | 2fba | 12218 | Same as Math2.2.Out |
| Math2.2.Oper | Operation (as Math2.1.Oper) | uint8 | 2fb7 | 12215 | Not applicable |
| Math2.2.Out | Output Value | float32 | 2fbb | 12219 | Set by Math2.2.Resolution |
| Math2.2.Resolution | Output Resolution | uint8 | 2fbf | 12223 | Not applicable |
| Math2.2.Select | Select Input 1 or Input 2 | bool | 2fbd | 12221 | Not applicable |
| Math2.2.Status | Status. 0 = Good (OK); 7 = Bad (Error) | uint8 | 2fbe | 12222 | Not applicable |
| Math2.2.Units | Output Units | string_t | 694a | 26954 | Not applicable |
| | | | | | |
| Math2.3.Fallback | Fallback strategy (as Math2.1.Fallback) | uint8 | 2fc9 | 12233 | Not applicable |
| Math2.3.FallbackVal | Fallback Value | float32 | 2fc5 | 12229 | Same as Math2.3.Out |
| Math2.3.HighLimit | Output High Limit | float32 | 2fc6 | 12230 | Same as Math2.3.Out |
| Math2.3.In1 | Input 1 Value | float32 | 2fc1 | 12225 | 0dp |
| Math2.3.In1Mul | Input 1 Scale | float32 | 2fc0 | 12224 | 1dp |
| Math2.3.In2 | Input 2 Value | float32 | 2fc3 | 12227 | 0dp |
| Math2.3.In2Mul | Input 2 Scale | float32 | 2fc2 | 12226 | 1dp |
| Math2.3.LowLimit | Output Low Limit | float32 | 2fc7 | 12231 | Same as Math2.3.Out |
| Math2.3.Oper | Operation (as Math2.1.Oper) | uint8 | 2fc4 | 12228 | Not applicable |
| Math2.3.Out | Output Value | float32 | 2fc8 | 12232 | Set by Math2.3.Resolution |
| Math2.3.Resolution | Output Resolution | uint8 | 2fcc | 12236 | Not applicable |
| Math2.3.Select | Select Between Input 1 and Input 2 | bool | 2fca | 12234 | Not applicable |
| Math2.3.Status | Status. 0 = Good (OK); 7 = Bad (Error) | uint8 | 2fcb | 12235 | Not applicable |
| Math2.3.Units | Output Units | string_t | 6950 | 26960 | Not applicable |
| | | | | | |
| Math2.4.Fallback | Fallback strategy (as Math2.1.Fallback) | uint8 | 2fd6 | 12246 | Not applicable |
| Math2.4.FallbackVal | Fallback Value | float32 | 2fd2 | 12242 | Same as Math2.4.Out |
| Math2.4.HighLimit | Output High Limit | float32 | 2fd3 | 12243 | Same as Math2.4.Out |
| Math2.4.In1 | Input 1 Value | float32 | 2fce | 12238 | 0dp |
| Math2.4.In1Mul | Input 1 Scale | float32 | 2fcd | 12237 | 1dp |
| Math2.4.In2 | Input 2 Value | float32 | 2fd0 | 12240 | 0dp |
| Math2.4.In2Mul | Input 2 Scale | float32 | 2fcf | 12239 | 1dp |
| Math2.4.LowLimit | Output Low Limit | float32 | 2fd4 | 12244 | Same as Math2.4.Out |
| Math2.4.Oper | Operation (as Math2.1.Oper) | uint8 | 2fd1 | 12241 | Not applicable |
| Math2.4.Out | Output Value | float32 | 2fd5 | 12245 | Set by Math2.4.Resolution |
| Math2.4.Resolution | Output Resolution | uint8 | 2fd9 | 12249 | Not applicable |
| Math2.4.Select | Select Between Input 1 and Input 2 | bool | 2fd7 | 12247 | Not applicable |
| Math2.4.Status | Status. 0 = Good (OK); 7 = Bad (Error) | uint8 | 2fd8 | 12248 | Not applicable |
| Math2.4.Units | Output Units | string_t | 6956 | 26966 | Not applicable |
| | | | | | |
| Math2.5.Fallback | Fallback strategy (as Math2.1.Fallback) | uint8 | 2fe3 | 12259 | Not applicable |
| Math2.5.FallbackVal | Fallback Value | float32 | 2fdf | 12255 | Same as Math2.5.Out |
| Math2.5.HighLimit | Output High Limit | float32 | 2fe0 | 12256 | Same as Math2.5.Out |
| Math2.5.In1 | Input 1 Value | float32 | 2fdb | 12251 | 0dp |
| Math2.5.In1Mul | Input 1 Scale | float32 | 2fda | 12250 | 1dp |
| Math2.5.In2 | Input 2 Value | float32 | 2fdd | 12253 | 0dp |
| Math2.5.In2Mul | Input 2 Scale | float32 | 2fdc | 12252 | 1dp |
| Math2.5.LowLimit | Output Low Limit | float32 | 2fe1 | 12257 | Same as Math2.5.Out |
| Math2.5.Oper | Operation (as Math2.1.Oper) | uint8 | 2fde | 12254 | Not applicable |
| Math2.5.Out | Output Value | float32 | 2fe2 | 12258 | Set by Math2.5.Resolution |
| Math2.5.Resolution | Output Resolution | uint8 | 2fe6 | 12262 | Not applicable |
| Math2.5.Select | Select Between Input 1 and Input 2 | bool | 2fe4 | 12260 | Not applicable |
| Math2.5.Status | Status. 0 = Good (OK); 7 = Bad (Error) | uint8 | 2fe5 | 12261 | Not applicable |
| Math2.5.Units | Output Units | string_t | 695c | 26972 | Not applicable |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| Math2.6.Fallback | Fallback strategy (as Math2.1.Fallback) | uint8 | 2ff0 | 12272 | Not applicable |
| Math2.6.FallbackVal | Fallback Value | float32 | 2fec | 12268 | Same as Math2.6.Out |
| Math2.6.HighLimit | Output High Limit | float32 | 2fed | 12269 | Same as Math2.6.Out |
| Math2.6.In1 | Input 1 Value | float32 | 2fe8 | 12264 | 0dp |
| Math2.6.In1Mul | Input 1 Scale | float32 | 2fe7 | 12263 | 1dp |
| Math2.6.In2 | Input 2 Value | float32 | 2fea | 12266 | 0dp |
| Math2.6.In2Mul | Input 2 Scale | float32 | 2fe9 | 12265 | 1dp |
| Math2.6.LowLimit | Output Low Limit | float32 | 2fee | 12270 | Same as Math2.6.Out |
| Math2.6.Oper | Operation (as Math2.1.Oper) | uint8 | 2feb | 12267 | Not applicable |
| Math2.6.Out | Output Value | float32 | 2fef | 12271 | Set by Math2.6.Resolution |
| Math2.6.Resolution | Output Resolution | uint8 | 2ff3 | 12275 | Not applicable |
| Math2.6.Select | Select Between Input 1 and Input 2 | bool | 2ff1 | 12273 | Not applicable |
| Math2.6.Status | Status. 0 = Good (OK); 7 = Bad (Error) | uint8 | 2ff2 | 12274 | Not applicable |
| Math2.6.Units | Output Units | string_t | 6962 | 26978 | Not applicable |
| Math2.7.Fallback | Fallback strategy (as Math2.1.Fallback) | uint8 | 2ffd | 12285 | Not applicable |
| Math2.7.FallbackVal | Fallback Value | float32 | 2ff9 | 12281 | Same as Math2.7.Out |
| Math2.7.HighLimit | Output High Limit | float32 | 2ffa | 12282 | Same as Math2.7.Out |
| Math2.7.In1 | Input 1 Value | float32 | 2ff5 | 12277 | 0dp |
| Math2.7.In1Mul | Input 1 Scale | float32 | 2ff4 | 12276 | 1dp |
| Math2.7.In2 | Input 2 Value | float32 | 2ff7 | 12279 | 0dp |
| Math2.7.In2Mul | Input 2 Scale | float32 | 2ff6 | 12278 | 1dp |
| Math2.7.LowLimit | Output Low Limit | float32 | 2ffb | 12283 | Same as Math2.7.Out |
| Math2.7.Oper | Operation (as Math2.1.Oper) | uint8 | 2ff8 | 12280 | Not applicable |
| Math2.7.Out | Output Value | float32 | 2ffc | 12284 | Set by Math2.7.Resolution |
| Math2.7.Resolution | Output Resolution | uint8 | 3000 | 12288 | Not applicable |
| Math2.7.Select | Select Between Input 1 and Input 2 | bool | 2ffe | 12286 | Not applicable |
| Math2.7.Status | Status. 0 = Good (OK); 7 = Bad (Error) | uint8 | 2fff | 12287 | Not applicable |
| Math2.7.Units | Output Units | string_t | 6968 | 26984 | Not applicable |
| Math2.8.Fallback | Fallback strategy (as Math2.1.Fallback) | uint8 | 300a | 12298 | Not applicable |
| Math2.8.FallbackVal | Fallback Value | float32 | 3006 | 12294 | Same as Math2.8.Out |
| Math2.8.HighLimit | Output High Limit | float32 | 3007 | 12295 | Same as Math2.8.Out |
| Math2.8.In1 | Input 1 Value | float32 | 3002 | 12290 | 0dp |
| Math2.8.In1Mul | Input 1 Scale | float32 | 3001 | 12289 | 1dp |
| Math2.8.In2 | Input 2 Value | float32 | 3004 | 12292 | 0dp |
| Math2.8.In2Mul | Input 2 Scale | float32 | 3003 | 12291 | 1dp |
| Math2.8.LowLimit | Output Low Limit | float32 | 3008 | 12296 | Same as Math2.8.Out |
| Math2.8.Oper | Operation (as Math2.1.Oper) | uint8 | 3005 | 12293 | Not applicable |
| Math2.8.Out | Output Value | float32 | 3009 | 12297 | Set by Math2.8.Resolution |
| Math2.8.Resolution | Output Resolution | uint8 | 300d | 12301 | Not applicable |
| Math2.8.Select | Select Between Input 1 and Input 2 | bool | 300b | 12299 | Not applicable |
| Math2.8.Status | Status. 0 = Good (OK); 7 = Bad (Error) | uint8 | 300c | 12300 | Not applicable |
| Math2.8.Units | Output Units | string_t | 696e | 26990 | Not applicable |
| Math2.9.Fallback | Fallback strategy (as Math2.1.Fallback) | uint8 | 3017 | 12311 | Not applicable |
| Math2.9.FallbackVal | Fallback Value | float32 | 3013 | 12307 | Same as Math2.9.Out |
| Math2.9.HighLimit | Output High Limit | float32 | 3014 | 12308 | Same as Math2.9.Out |
| Math2.9.In1 | Input 1 Value | float32 | 300f | 12303 | 0dp |
| Math2.9.In1Mul | Input 1 Scale | float32 | 300e | 12302 | 1dp |
| Math2.9.In2 | Input 2 Value | float32 | 3011 | 12305 | 0dp |
| Math2.9.In2Mul | Input 2 Scale | float32 | 3010 | 12304 | 1dp |
| Math2.9.LowLimit | Output Low Limit | float32 | 3015 | 12309 | Same as Math2.9.Out |
| Math2.9.Oper | Operation (as Math2.1.Oper) | uint8 | 3012 | 12306 | Not applicable |
| Math2.9.Out | Output Value | float32 | 3016 | 12310 | Set by Math2.9.Resolution |
| Math2.9.Resolution | Output Resolution | uint8 | 301a | 12314 | Not applicable |
| Math2.9.Select | Select Between Input 1 and Input 2 | bool | 3018 | 12312 | Not applicable |
| Math2.9.Status | Status. 0 = Good (OK); 7 = Bad (Error) | uint8 | 3019 | 12313 | Not applicable |
| Math2.9.Units | Output Units | string_t | 6974 | 26996 | Not applicable |
| Math2.10.Fallback | Fallback strategy (as Math2.1.Fallback) | uint8 | 3024 | 12324 | Not applicable |
| Math2.10.FallbackVal | Fallback Value | float32 | 3020 | 12320 | Same as Math2.10.Out |
| Math2.10.HighLimit | Output High Limit | float32 | 3021 | 12321 | Same as Math2.10.Out |
| Math2.10.In1 | Input 1 Value | float32 | 301c | 12316 | 0dp |
| Math2.10.In1Mul | Input 1 Scale | float32 | 301b | 12315 | 1dp |
| Math2.10.In2 | Input 2 Value | float32 | 301e | 12318 | 0dp |
| Math2.10.In2Mul | Input 2 Scale | float32 | 301d | 12317 | 1dp |
| Math2.10.LowLimit | Output Low Limit | float32 | 3022 | 12322 | Same as Math2.10.Out |
| Math2.10.Oper | Operation (as Math2.1.Oper) | uint8 | 301f | 12319 | Not applicable |
| Math2.10.Out | Output Value | float32 | 3023 | 12323 | Set by Math2.10.Resolution |
| Math2.10.Resolution | Output Resolution | uint8 | 3027 | 12327 | Not applicable |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| Math2.10.Select | Select Between Input 1 and Input 2 | bool | 3025 | 12325 | Not applicable |
| Math2.10.Status | Status. 0 = Good (OK); 7 = Bad (Error) | uint8 | 3026 | 12326 | Not applicable |
| Math2.10.Units | Output Units | string_t | 697a | 27002 | Not applicable |
| Math2.11.Fallback | Fallback strategy (as Math2.1.Fallback) | uint8 | 3031 | 12337 | Not applicable |
| Math2.11.FallbackVal | Fallback Value | float32 | 302d | 12333 | Same as Math2.11.Out |
| Math2.11.HighLimit | Output High Limit | float32 | 302e | 12334 | Same as Math2.11.Out |
| Math2.11.In1 | Input 1 Value | float32 | 3029 | 12329 | 0dp |
| Math2.11.In1Mul | Input 1 Scale | float32 | 3028 | 12328 | 1dp |
| Math2.11.In2 | Input 2 Value | float32 | 302b | 12331 | 0dp |
| Math2.11.In2Mul | Input 2 Scale | float32 | 302a | 12330 | 1dp |
| Math2.11.LowLimit | Output Low Limit | float32 | 302f | 12335 | Same as Math2.11.Out |
| Math2.11.Oper | Operation (as Math2.1.Oper) | uint8 | 302c | 12332 | Not applicable |
| Math2.11.Out | Output Value | float32 | 3030 | 12336 | Set by Math2.11.Resolution |
| Math2.11.Resolution | Output Resolution | uint8 | 3034 | 12340 | Not applicable |
| Math2.11.Select | Select Between Input 1 and Input 2 | bool | 3032 | 12338 | Not applicable |
| Math2.11.Status | Status. 0 = Good (OK); 7 = Bad (Error) | uint8 | 3033 | 12339 | Not applicable |
| Math2.11.Units | Output Units | string_t | 6980 | 27008 | Not applicable |
| Math2.12.Fallback | Fallback strategy (as Math2.1.Fallback) | uint8 | 303e | 12350 | Not applicable |
| Math2.12.FallbackVal | Fallback Value | float32 | 303a | 12346 | Same as Math2.12.Out |
| Math2.12.HighLimit | Output High Limit | float32 | 303b | 12347 | Same as Math2.12.Out |
| Math2.12.In1 | Input 1 Value | float32 | 3036 | 12342 | 0dp |
| Math2.12.In1Mul | Input 1 Scale | float32 | 3035 | 12341 | 1dp |
| Math2.12.In2 | Input 2 Value | float32 | 3038 | 12344 | 0dp |
| Math2.12.In2Mul | Input 2 Scale | float32 | 3037 | 12343 | 1dp |
| Math2.12.LowLimit | Output Low Limit | float32 | 303c | 12348 | Same as Math2.12.Out |
| Math2.12.Oper | Operation (as Math2.1.Oper) | uint8 | 3039 | 12345 | Not applicable |
| Math2.12.Out | Output Value | float32 | 303d | 12349 | Set by Math2.12.Resolution |
| Math2.12.Resolution | Output Resolution | uint8 | 3041 | 12353 | Not applicable |
| Math2.12.Select | Select Between Input 1 and Input 2 | bool | 303f | 12351 | Not applicable |
| Math2.12.Status | Status. 0 = Good (OK); 7 = Bad (Error) | uint8 | 3040 | 12352 | Not applicable |
| Math2.12.Units | Output Units | string_t | 6986 | 27014 | Not applicable |
| ModbusMaster.1.Data.AlarmStatus | Alarm status (0 = No alarms; 1 = one or more alarms active) | uint8 | 7dbb | 32187 | Not applicable |
| ModbusMaster.1.Data.BitPosition | Bit position of the bit of interest in a 16 bit data type | uint8 | 7d4f | 32079 | Not applicable |
| ModbusMaster.1.Data.ChanAlarmStatus | Channel alarm status<br>0 = Off 1 = Active 2 = Safe Nackd 3 = Active Nackd | uint8 | 7ddb | 32219 | Not applicable |
| ModbusMaster.1.Data.DataType | Data type of the data being read/written<br>0 = Real 1 = DINT 2 = INT 3 = Byte<br>4 = UDINT 5 = UINT 6 = UBYTE 8 = Real (Swap)<br>9 = DINT (Swap) 10 = UDINT (Swap) 11 = BIT | uint8 | 7c06 | 31750 | Not applicable |
| ModbusMaster.1.Data.Descriptor | Description for this data item | string_t | 6687 | 26247 | Not applicable |
| ModbusMaster.1.Data.Digital | Digital status (0 = Off, 1 = On) | bool | 7e1b | 32283 | Not applicable |
| ModbusMaster.1.Data.FallBackValue | Fall back value to be writen to the slave device | float32 | 7c7e | 31870 | 2dp |
| ModbusMaster.1.Data.FunctionCode | The modbus function codeuint8<br>1 = Read coil 2 = Read discrete 3 = Read holding<br>4 = Read input 5 = Write coil 6 = Write single<br>16 = Write multiple | 7be8 | | 31720 | Not applicable |
| ModbusMaster.1.Data.ModbusAddress | Modbus register address of the data to be read/written | float32 | 7b8c | 31628 | 0dp |
| ModbusMaster.1.Data.Mode | Auto Manual mode selection (0 = Auto; 1 = Manual) | uint8 | 7d9b | 32155 | Not applicable |
| ModbusMaster.1.Data.Number | Used for multiple instance parameters | uint8 | 7d13 | 32019 | Not applicable |
| ModbusMaster.1.Data.ParameterList | Parameter list for a specific slave device | uint8 | 7cf5 | 31989 | Not applicable |
| ModbusMaster.1.Data.Priority | Frequency at which the data is read/written<br>0 = High 1 = Medium 2 = Low 3 = Acyclic | uint8 | 7c24 | 31780 | Not applicable |
| ModbusMaster.1.Data.PV | Process value recieved from slave device | float32 | 7b32 | 31538 | 2dp |
| ModbusMaster.1.Data.Scaling | Scaling in decimal places for non floating point data types | uint8 | 7d31 | 32049 | Not applicable |
| ModbusMaster.1.Data.Send | 1 = send the write value to the slave | bool | 7cb9 | 31929 | Not applicable |
| ModbusMaster.1.Data.Set | Sets a digital value (1 = on; 0 = off) | bool | 7dfb | 32251 | Not applicable |
| ModbusMaster.1.Data.SlaveDevice | Slave device to communicate with. | uint8 | 7b14 | 31508 | Not applicable |
| ModbusMaster.1.Data.Status | Transaction status<br>0 = Success 1 = Illegal function 2 = Ilegal address<br>6 = Slave busy 8 = Parity error 9 = Bad sub<br>10 = Bad gateway 11 = No response 12 = Idle<br>13 = Pending 14 = Timeout 15 = Unknown host<br>16 = Connect fail 17 = No sockets 18 = Loopback fail<br>19 = Login fail 20 = Unknown error 22 = Write fail<br>23 = Master reject | uint8 | 7cd7 | 31959 | Not applicable |
| ModbusMaster.1.Data.Value | The value to be written to the slave device | float32 | 7c42 | 31810 | 2dp |
| ModbusMaster.2.Data.AlarmStatus | Alarm status (as for Modbus Master.1) | uint8 | 7dbc | 32188 | Not applicable |
| ModbusMaster.2.Data.BitPosition | Bit position of the bit of interest in a 16 bit data type | uint8 | 7d50 | 32080 | Not applicable |
| ModbusMaster.2.Data.ChanAlarmStatus | Channel alarm status (as for Modbus Master.1) | uint8 | 7ddc | 32220 | Not applicable |
| ModbusMaster.2.Data.DataType | Type of data being read/written (as for Modbus Master.1) | uint8 | 7c07 | 31751 | Not applicable |
| ModbusMaster.2.Data.Descriptor | Description for this data item | string_t | 669c | 26268 | Not applicable |
| ModbusMaster.2.Data.Digital | Digital status (0 = Off, 1 = On) | bool | 7e1c | 32284 | Not applicable |
| ModbusMaster.2.Data.FallBackValue | Fall back value to be writen to the slave device | float32 | 7c80 | 31872 | 2dp |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| ModbusMaster.2.Data.FunctionCode | The modbus function code (as for Modbus Master.1) | uint8 | 7be9 | 31721 | Not applicable |
| ModbusMaster.2.Data.ModbusAddress | Modbus register address of the data to be read/written | float32 | 7b8e | 31630 | 0dp |
| ModbusMaster.2.Data.Mode | Auto Manual mode selection (0 = Auto; 1 = Manual) | uint8 | 7d9c | 32156 | Not applicable |
| ModbusMaster.2.Data.Number | Used for multiple instance parameters | uint8 | 7d14 | 32020 | Not applicable |
| ModbusMaster.2.Data.ParameterList | Parameter list for a specific slave device | uint8 | 7cf6 | 31990 | Not applicable |
| ModbusMaster.2.Data.Priority | Read/Write frequency (as for Modbus Master.1) | uint8 | 7c25 | 31781 | Not applicable |
| ModbusMaster.2.Data.PV | Process value recieved from slave device | float32 | 7b34 | 31540 | 2dp |
| ModbusMaster.2.Data.Scaling | Scaling in decimal places for non floating point data types | uint8 | 7d32 | 32050 | Not applicable |
| ModbusMaster.2.Data.Send | 1 = send the write value to the slave | bool | 7cba | 31930 | Not applicable |
| ModbusMaster.2.Data.Set | Sets a digital value (1 = on; 0 = off) | bool | 7dfc | 32252 | Not applicable |
| ModbusMaster.2.Data.SlaveDevice | Slave device to communicate with. | uint8 | 7b15 | 31509 | Not applicable |
| ModbusMaster.2.Data.Status | Transaction status (as for Modbus Master.1) | uint8 | 7cd8 | 31960 | Not applicable |
| ModbusMaster.2.Data.Value | The value to be written to the slave device | float32 | 7c44 | 31812 | 2dp |
| | | | | | |
| ModbusMaster.3.Data.AlarmStatus | Alarm status (as for Modbus Master.1) | uint8 | 7dbd | 32189 | Not applicable |
| ModbusMaster.3.Data.BitPosition | Bit position of the bit of interest in a 16 bit data type | uint8 | 7d51 | 32081 | Not applicable |
| ModbusMaster.3.Data.ChanAlarmStatus | Channel alarm status (as for Modbus Master.1) | uint8 | 7ddd | 32221 | Not applicable |
| ModbusMaster.3.Data.DataType | Type of data being read/written (as for Modbus Master.1) | uint8 | 7c08 | 31752 | Not applicable |
| ModbusMaster.3.Data.Descriptor | Description for this data item | string_t | 66b1 | 26289 | Not applicable |
| ModbusMaster.3.Data.Digital | Digital status (0 = Off, 1 = On) | bool | 7e1d | 32285 | Not applicable |
| ModbusMaster.3.Data.FallBackValue | Fall back value to be writen to the slave device | float32 | 7c82 | 31874 | 2dp |
| ModbusMaster.3.Data.FunctionCode | The modbus function code (as for Modbus Master.1) | uint8 | 7bea | 31722 | Not applicable |
| ModbusMaster.3.Data.ModbusAddress | Modbus register address of the data to be read/written | float32 | 7b90 | 31632 | 0dp |
| ModbusMaster.3.Data.Mode | Auto Manual mode selection (0 = Auto; 1 = Manual) | uint8 | 7d9d | 32157 | Not applicable |
| ModbusMaster.3.Data.Number | Used for multiple instance parameters | uint8 | 7d15 | 32021 | Not applicable |
| ModbusMaster.3.Data.ParameterList | Parameter list for a specific slave device | uint8 | 7cf7 | 31991 | Not applicable |
| ModbusMaster.3.Data.Priority | Read/Write frequency (as for Modbus Master.1) | uint8 | 7c26 | 31782 | Not applicable |
| ModbusMaster.3.Data.PV | Process value recieved from slave device | float32 | 7b36 | 31542 | 2dp |
| ModbusMaster.3.Data.Scaling | Scaling in decimal places for non floating point data types | uint8 | 7d33 | 32051 | Not applicable |
| ModbusMaster.3.Data.Send | 1 = send the write value to the slave | bool | 7cbb | 31931 | Not applicable |
| ModbusMaster.3.Data.Set | Sets a digital value (1 = on; 0 = off) | bool | 7dfd | 32253 | Not applicable |
| ModbusMaster.3.Data.SlaveDevice | Slave device to communicate with. | uint8 | 7b16 | 31510 | Not applicable |
| ModbusMaster.3.Data.Status | Transaction status (as for Modbus Master.1 | uint8 | 7cd9 | 31961 | Not applicable |
| ModbusMaster.3.Data.Value | The value to be written to the slave device | float32 | 7c46 | 31814 | 2dp |
| | | | | | |
| ModbusMaster.4.Data.AlarmStatus | Alarm status (as for Modbus Master.1) | uint8 | 7dbe | 32190 | Not applicable |
| ModbusMaster.4.Data.BitPosition | Bit position of the bit of interest in a 16 bit data type | uint8 | 7d52 | 32082 | Not applicable |
| ModbusMaster.4.Data.ChanAlarmStatus | Channel alarm status (as for Modbus Master.1) | uint8 | 7dde | 32222 | Not applicable |
| ModbusMaster.4.Data.DataType | Type of data being read/written (as for Modbus Master.1) | uint8 | 7c09 | 31753 | Not applicable |
| ModbusMaster.4.Data.Descriptor | Description for this data item | string_t | 66c6 | 26310 | Not applicable |
| ModbusMaster.4.Data.Digital | Digital status (0 = Off, 1 = On) | bool | 7e1e | 32286 | Not applicable |
| ModbusMaster.4.Data.FallBackValue | Fall back value to be writen to the slave device | float32 | 7c84 | 31876 | 2dp |
| ModbusMaster.4.Data.FunctionCode | The modbus function code (as for Modbus Master.1) | uint8 | 7beb | 31723 | Not applicable |
| ModbusMaster.4.Data.ModbusAddress | Modbus register address of the data to be read/written | float32 | 7b92 | 31634 | 0dp |
| ModbusMaster.4.Data.Mode | Auto Manual mode selection (0 = Auto; 1 = Manual) | uint8 | 7d9e | 32158 | Not applicable |
| ModbusMaster.4.Data.Number | Used for multiple instance parameters | uint8 | 7d16 | 32022 | Not applicable |
| ModbusMaster.4.Data.ParameterList | Parameter list for a specific slave device | uint8 | 7cf8 | 31992 | Not applicable |
| ModbusMaster.4.Data.Priority | Read/Write frequency (as for Modbus Master.1) | uint8 | 7c27 | 31783 | Not applicable |
| ModbusMaster.4.Data.PV | Process value recieved from slave device | float32 | 7b38 | 31544 | 2dp |
| ModbusMaster.4.Data.Scaling | Scaling in decimal places for non floating point data types | uint8 | 7d34 | 32052 | Not applicable |
| ModbusMaster.4.Data.Send | 1 = send the write value to the slave | bool | 7cbc | 31932 | Not applicable |
| ModbusMaster.4.Data.Set | Sets a digital value (1 = on; 0 = off) | bool | 7dfe | 32254 | Not applicable |
| ModbusMaster.4.Data.SlaveDevice | Slave device to communicate with. | uint8 | 7b17 | 31511 | Not applicable |
| ModbusMaster.4.Data.Status | Transaction status (as for Modbus Master.1 | uint8 | 7cda | 31962 | Not applicable |
| ModbusMaster.4.Data.Value | The value to be written to the slave device | float32 | 7c48 | 31816 | 2dp |
| | | | | | |
| ModbusMaster.5.Data.AlarmStatus | Alarm status (as for Modbus Master.1) | uint8 | 7dbf | 32191 | Not applicable |
| ModbusMaster.5.Data.BitPosition | Bit position of the bit of interest in a 16 bit data type | uint8 | 7d53 | 32083 | Not applicable |
| ModbusMaster.5.Data.ChanAlarmStatus | Channel alarm status (as for Modbus Master.1) | uint8 | 7ddf | 32223 | Not applicable |
| ModbusMaster.5.Data.DataType | Type of data being read/written (as for Modbus Master.1) | uint8 | 7c0a | 31754 | Not applicable |
| ModbusMaster.5.Data.Descriptor | Description for this data item | string_t | 66db | 26331 | Not applicable |
| ModbusMaster.5.Data.Digital | Digital status (0 = Off, 1 = On) | bool | 7e1f | 32287 | Not applicable |
| ModbusMaster.5.Data.FallBackValue | Fall back value to be writen to the slave device | float32 | 7c86 | 31878 | 2dp |
| ModbusMaster.5.Data.FunctionCode | The modbus function code (as for Modbus Master.1) | uint8 | 7bec | 31724 | Not applicable |
| ModbusMaster.5.Data.ModbusAddress | Modbus register address of the data to be read/written | float32 | 7b94 | 31636 | 0dp |
| ModbusMaster.5.Data.Mode | Auto Manual mode selection (0 = Auto; 1 = Manual) | uint8 | 7d9f | 32159 | Not applicable |
| ModbusMaster.5.Data.Number | Used for multiple instance parameters | uint8 | 7d17 | 32023 | Not applicable |
| ModbusMaster.5.Data.ParameterList | Parameter list for a specific slave device | uint8 | 7cf9 | 31993 | Not applicable |
| ModbusMaster.5.Data.Priority | Read/Write frequency (as for Modbus Master.1) | uint8 | 7c28 | 31784 | Not applicable |
| ModbusMaster.5.Data.PV | Process value recieved from slave device | float32 | 7b3a | 31546 | 2dp |
| ModbusMaster.5.Data.Scaling | Scaling in decimal places for non floating point data types | uint8 | 7d35 | 32053 | Not applicable |
| ModbusMaster.5.Data.Send | 1 = send the write value to the slave | bool | 7cbd | 31933 | Not applicable |
| ModbusMaster.5.Data.Set | Sets a digital value (1 = on; 0 = off) | bool | 7dff | 32255 | Not applicable |
| ModbusMaster.5.Data.SlaveDevice | Slave device to communicate with. | uint8 | 7b18 | 31512 | Not applicable |
| ModbusMaster.5.Data.Status | Transaction status (as for Modbus Master.1 | uint8 | 7cdb | 31963 | Not applicable |
| ModbusMaster.5.Data.Value | The value to be written to the slave device | float32 | 7c4a | 31818 | 2dp |
| | | | | | |
| ModbusMaster.6.Data.AlarmStatus | Alarm status (as for Modbus Master.1) | uint8 | 7dc0 | 32192 | Not applicable |
| ModbusMaster.6.Data.BitPosition | Bit position of the bit of interest in a 16 bit data type | uint8 | 7d54 | 32084 | Not applicable |
| ModbusMaster.6.Data.ChanAlarmStatus | Channel alarm status (as for Modbus Master.1) | uint8 | 7de0 | 32224 | Not applicable |
| ModbusMaster.6.Data.DataType | Type of data being read/written (as for Modbus Master.1) | uint8 | 7c0b | 31755 | Not applicable |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| ModbusMaster.6.Data.Descriptor | Description for this data item | string_t | 66f0 | 26352 | Not applicable |
| ModbusMaster.6.Data.Digital | Digital status (0 = Off, 1 = On) | bool | 7e20 | 32288 | Not applicable |
| ModbusMaster.6.Data.FallBackValue | Fall back value to be writen to the slave device | float32 | 7c88 | 31880 | 2dp |
| ModbusMaster.6.Data.FunctionCode | The modbus function code (as for Modbus Master.1) | uint8 | 7bed | 31725 | Not applicable |
| ModbusMaster.6.Data.ModbusAddress | Modbus register address of the data to be read/written | float32 | 7b96 | 31638 | 0dp |
| ModbusMaster.6.Data.Mode | Auto Manual mode selection (0 = Auto; 1 = Manual) | uint8 | 7da0 | 32160 | Not applicable |
| ModbusMaster.6.Data.Number | Used for multiple instance parameters | uint8 | 7d18 | 32024 | Not applicable |
| ModbusMaster.6.Data.ParameterList | Parameter list for a specific slave device | uint8 | 7cfa | 31994 | Not applicable |
| ModbusMaster.6.Data.Priority | Read/Write frequency (as for Modbus Master.1) | uint8 | 7c29 | 31785 | Not applicable |
| ModbusMaster.6.Data.PV | Process value recieved from slave device | float32 | 7b3c | 31548 | 2dp |
| ModbusMaster.6.Data.Scaling | Scaling in decimal places for non floating point data types | uint8 | 7d36 | 32054 | Not applicable |
| ModbusMaster.6.Data.Send | 1 = send the write value to the slave device | bool | 7cbe | 31934 | Not applicable |
| ModbusMaster.6.Data.Set | Sets a digital value (1 = on; 0 = off) | bool | 7e00 | 32256 | Not applicable |
| ModbusMaster.6.Data.SlaveDevice | Slave device to communicate with. | uint8 | 7b19 | 31513 | Not applicable |
| ModbusMaster.6.Data.Status | Transaction status (as for Modbus Master.1 | uint8 | 7cdc | 31964 | Not applicable |
| ModbusMaster.6.Data.Value | The value to be written to the slave device | float32 | 7c4c | 31820 | 2dp |
| ModbusMaster.7.Data.AlarmStatus | Alarm status (as for Modbus Master.1) | uint8 | 7dc1 | 32193 | Not applicable |
| ModbusMaster.7.Data.BitPosition | Bit position of the bit of interest in a 16 bit data type | uint8 | 7d55 | 32085 | Not applicable |
| ModbusMaster.7.Data.ChanAlarmStatus | Channel alarm status (as for Modbus Master.1) | uint8 | 7de1 | 32225 | Not applicable |
| ModbusMaster.7.Data.DataType | Type of data being read/written (as for Modbus Master.1) | uint8 | 7c0c | 31756 | Not applicable |
| ModbusMaster.7.Data.Descriptor | Description for this data item | string_t | 6705 | 26373 | Not applicable |
| ModbusMaster.7.Data.Digital | Digital status (0 = Off, 1 = On) | bool | 7e21 | 32289 | Not applicable |
| ModbusMaster.7.Data.FallBackValue | Fall back value to be writen to the slave device | float32 | 7c8a | 31882 | 2dp |
| ModbusMaster.7.Data.FunctionCode | The modbus function code (as for Modbus Master.1) | uint8 | 7bee | 31726 | Not applicable |
| ModbusMaster.7.Data.ModbusAddress | Modbus register address of the data to be read/written | float32 | 7b98 | 31640 | 0dp |
| ModbusMaster.7.Data.Mode | Auto Manual mode selection (0 = Auto; 1 = Manual) | uint8 | 7da1 | 32161 | Not applicable |
| ModbusMaster.7.Data.Number | Used for multiple instance parameters | uint8 | 7d19 | 32025 | Not applicable |
| ModbusMaster.7.Data.ParameterList | Parameter list for a specific slave device | uint8 | 7cfb | 31995 | Not applicable |
| ModbusMaster.7.Data.Priority | Read/Write frequency (as for Modbus Master.1) | uint8 | 7c2a | 31786 | Not applicable |
| ModbusMaster.7.Data.PV | Process value recieved from slave device | float32 | 7b3e | 31550 | 2dp |
| ModbusMaster.7.Data.Scaling | Scaling in decimal places for non floating point data types | uint8 | 7d37 | 32055 | Not applicable |
| ModbusMaster.7.Data.Send | 1 = send the write value to the slave | bool | 7cbf | 31935 | Not applicable |
| ModbusMaster.7.Data.Set | Sets a digital value (1 = on; 0 = off) | bool | 7e01 | 32257 | Not applicable |
| ModbusMaster.7.Data.SlaveDevice | Slave device to communicate with. | uint8 | 7b1a | 31514 | Not applicable |
| ModbusMaster.7.Data.Status | Transaction status (as for Modbus Master.1 | uint8 | 7cdd | 31965 | Not applicable |
| ModbusMaster.7.Data.Value | The value to be written to the slave device | float32 | 7c4e | 31822 | 2dp |
| ModbusMaster.8.Data.AlarmStatus | Alarm status (as for Modbus Master.1) | uint8 | 7dc2 | 32194 | Not applicable |
| ModbusMaster.8.Data.BitPosition | Bit position of the bit of interest in a 16 bit data type | uint8 | 7d56 | 32086 | Not applicable |
| ModbusMaster.8.Data.ChanAlarmStatus | Channel alarm status (as for Modbus Master.1) | uint8 | 7de2 | 32226 | Not applicable |
| ModbusMaster.8.Data.DataType | Type of data being read/written (as for Modbus Master.1) | uint8 | 7c0d | 31757 | Not applicable |
| ModbusMaster.8.Data.Descriptor | Description for this data item | string_t | 671a | 26394 | Not applicable |
| ModbusMaster.8.Data.Digital | Digital status (0 = Off, 1 = On) | bool | 7e22 | 32290 | Not applicable |
| ModbusMaster.8.Data.FallBackValue | Fall back value to be writen to the slave device | float32 | 7c8c | 31884 | 2dp |
| ModbusMaster.8.Data.FunctionCode | The modbus function code (as for Modbus Master.1) | uint8 | 7bef | 31727 | Not applicable |
| ModbusMaster.8.Data.ModbusAddress | Modbus register address of the data to be read/written | float32 | 7b9a | 31642 | 0dp |
| ModbusMaster.8.Data.Mode | Auto Manual mode selection (0 = Auto; 1 = Manual) | uint8 | 7da2 | 32162 | Not applicable |
| ModbusMaster.8.Data.Number | Used for multiple instance parameters | uint8 | 7d1a | 32026 | Not applicable |
| ModbusMaster.8.Data.ParameterList | Parameter list for a specific slave device | uint8 | 7cfc | 31996 | Not applicable |
| ModbusMaster.8.Data.Priority | Read/Write frequency (as for Modbus Master.1) | uint8 | 7c2b | 31787 | Not applicable |
| ModbusMaster.8.Data.PV | Process value recieved from slave device | float32 | 7b40 | 31552 | 2dp |
| ModbusMaster.8.Data.Scaling | Scaling in decimal places for non floating point data types | uint8 | 7d38 | 32056 | Not applicable |
| ModbusMaster.8.Data.Send | 1 = send the write value to the slave | bool | 7cc0 | 31936 | Not applicable |
| ModbusMaster.8.Data.Set | Sets a digital value (1 = on; 0 = off) | bool | 7e02 | 32258 | Not applicable |
| ModbusMaster.8.Data.SlaveDevice | Slave device to communicate with. | uint8 | 7b1b | 31515 | Not applicable |
| ModbusMaster.8.Data.Status | Transaction status (as for Modbus Master.1 | uint8 | 7cde | 31966 | Not applicable |
| ModbusMaster.8.Data.Value | The value to be written to the slave device | float32 | 7c50 | 31824 | 2dp |
| ModbusMaster.9.Data.AlarmStatus | Alarm status (as for Modbus Master.1) | uint8 | 7dc3 | 32195 | Not applicable |
| ModbusMaster.9.Data.BitPosition | Bit position of the bit of interest in a 16 bit data type | uint8 | 7d57 | 32087 | Not applicable |
| ModbusMaster.9.Data.ChanAlarmStatus | Channel alarm status (as for Modbus Master.1) | uint8 | 7de3 | 32227 | Not applicable |
| ModbusMaster.9.Data.DataType | Type of data being read/written (as for Modbus Master.1) | uint8 | 7c0e | 31758 | Not applicable |
| ModbusMaster.9.Data.Descriptor | Description for this data item | string_t | 672f | 26415 | Not applicable |
| ModbusMaster.9.Data.Digital | Digital status (0 = Off, 1 = On) | bool | 7e23 | 32291 | Not applicable |
| ModbusMaster.9.Data.FallBackValue | Fall back value to be writen to the slave device | float32 | 7c8e | 31886 | 2dp |
| ModbusMaster.9.Data.FunctionCode | The modbus function code (as for Modbus Master.1) | uint8 | 7bf0 | 31728 | Not applicable |
| ModbusMaster.9.Data.ModbusAddress | Modbus register address of the data to be read/written | float32 | 7b9c | 31644 | 0dp |
| ModbusMaster.9.Data.Mode | Auto Manual mode selection (0 = Auto; 1 = Manual) | uint8 | 7da3 | 32163 | Not applicable |
| ModbusMaster.9.Data.Number | Used for multiple instance parameters | uint8 | 7d1b | 32027 | Not applicable |
| ModbusMaster.9.Data.ParameterList | Parameter list for a specific slave device | uint8 | 7cfd | 31997 | Not applicable |
| ModbusMaster.9.Data.Priority | FRead/Write frequency (as for Modbus Master.1) | uint8 | 7c2c | 31788 | Not applicable |
| ModbusMaster.9.Data.PV | Process value recieved from slave device | float32 | 7b42 | 31554 | 2dp |
| ModbusMaster.9.Data.Scaling | Scaling in decimal places for non floating point data types | uint8 | 7d39 | 32057 | Not applicable |
| ModbusMaster.9.Data.Send | 1 = send the write value to the slave | bool | 7cc1 | 31937 | Not applicable |
| ModbusMaster.9.Data.Set | Sets a digital value (1 = on; 0 = off) | bool | 7e03 | 32259 | Not applicable |
| ModbusMaster.9.Data.SlaveDevice | Slave device to communicate with. | uint8 | 7b1c | 31516 | Not applicable |
| ModbusMaster.9.Data.Status | Transaction status (as for Modbus Master.1 | uint8 | 7cdf | 31967 | Not applicable |
| ModbusMaster.9.Data.Value | The value to be written to the slave device | float32 | 7c52 | 31826 | 2dp |
| ModbusMaster.10.Data.AlarmStatus | Alarm status (as for Modbus Master.1) | uint8 | 7dc4 | 32196 | Not applicable |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| ModbusMaster.10.Data.BitPosition | Bit position of the bit of interest in a 16 bit data type | uint8 | 7d58 | 32088 | Not applicable |
| ModbusMaster.10.Data.ChanAlarmStatus | Channel alarm status (as for Modbus Master.1) | uint8 | 7de4 | 32228 | Not applicable |
| ModbusMaster.10.Data.DataType | Type of data being read/written (as for Modbus Master.1) | uint8 | 7c0f | 31759 | Not applicable |
| ModbusMaster.10.Data.Descriptor | Description for this data item | string_t | 6744 | 26436 | Not applicable |
| ModbusMaster.10.Data.Digital | Digital status (0 = Off, 1 = On) | bool | 7e24 | 32292 | Not applicable |
| ModbusMaster.10.Data.FallBackValue | Fall back value to be writen to the slave device | float32 | 7c90 | 31888 | 2dp |
| ModbusMaster.10.Data.FunctionCode | The modbus function code (as for Modbus Master.1) | uint8 | 7bf1 | 31729 | Not applicable |
| ModbusMaster.10.Data.ModbusAddress | Modbus register address of the data to be read/written | float32 | 7b9e | 31646 | 0dp |
| ModbusMaster.10.Data.Mode | Auto Manual mode selection (0 = Auto; 1 = Manual) | uint8 | 7da4 | 32164 | Not applicable |
| ModbusMaster.10.Data.Number | Used for multiple instance parameters | uint8 | 7d1c | 32028 | Not applicable |
| ModbusMaster.10.Data.ParameterList | Parameter list for a specific slave device | uint8 | 7cfe | 31998 | Not applicable |
| ModbusMaster.10.Data.Priority | Read/Write frequency (as for Modbus Master.1) | uint8 | 7c2d | 31789 | Not applicable |
| ModbusMaster.10.Data.PV | Process value recieved from slave device | float32 | 7b44 | 31556 | 2dp |
| ModbusMaster.10.Data.Scaling | Scaling in decimal places for non floating point data types | uint8 | 7d3a | 32058 | Not applicable |
| ModbusMaster.10.Data.Send | 1 = send the write value to the slave | bool | 7cc2 | 31938 | Not applicable |
| ModbusMaster.10.Data.Set | Sets a digital value (1 = on; 0 = off) | bool | 7e04 | 32260 | Not applicable |
| ModbusMaster.10.Data.SlaveDevice | Slave device to communicate with. | uint8 | 7b1d | 31517 | Not applicable |
| ModbusMaster.10.Data.Status | Transaction status (as for Modbus Master.1 | uint8 | 7ce0 | 31968 | Not applicable |
| ModbusMaster.10.Data.Value | The value to be written to the slave device | float32 | 7c54 | 31828 | 2dp |
| ModbusMaster.11.Data.AlarmStatus | Alarm status (as for Modbus Master.1) | uint8 | 7dc5 | 32197 | Not applicable |
| ModbusMaster.11.Data.BitPosition | Bit position of the bit of interest in a 16 bit data type | uint8 | 7d59 | 32089 | Not applicable |
| ModbusMaster.11.Data.ChanAlarmStatus | Channel alarm status (as for Modbus Master.1) | uint8 | 7de5 | 32229 | Not applicable |
| ModbusMaster.11.Data.DataType | Type of data being read/written (as for Modbus Master.1) | uint8 | 7c10 | 31760 | Not applicable |
| ModbusMaster.11.Data.Descriptor | Description for this data item | string_t | 6759 | 26457 | Not applicable |
| ModbusMaster.11.Data.Digital | Digital status (0 = Off, 1 = On) | bool | 7e25 | 32293 | Not applicable |
| ModbusMaster.11.Data.FallBackValue | Fall back value to be writen to the slave device | float32 | 7c92 | 31890 | 2dp |
| ModbusMaster.11.Data.FunctionCode | The modbus function code (as for Modbus Master.1) | uint8 | 7bf2 | 31730 | Not applicable |
| ModbusMaster.11.Data.ModbusAddress | Modbus register address of the data to be read/written | float32 | 7bc0 | 31680 | 0dp |
| ModbusMaster.11.Data.Mode | Auto Manual mode selection (0 = Auto; 1 = Manual) | uint8 | 7da5 | 32165 | Not applicable |
| ModbusMaster.11.Data.Number | Used for multiple instance parameters | uint8 | 7d1d | 32029 | Not applicable |
| ModbusMaster.11.Data.ParameterList | Parameter list for a specific slave device | uint8 | 7cff | 31999 | Not applicable |
| ModbusMaster.11.Data.Priority | Read/Write frequency (as for Modbus Master.1) | uint8 | 7c2e | 31790 | Not applicable |
| ModbusMaster.11.Data.PV | Process value recieved from slave device | float32 | 7b46 | 31558 | 2dp |
| ModbusMaster.11.Data.Scaling | Scaling in decimal places for non floating point data types | uint8 | 7d3b | 32059 | Not applicable |
| ModbusMaster.11.Data.Send | 1 = send the write value to the slave | bool | 7cc3 | 31939 | Not applicable |
| ModbusMaster.11.Data.Set | Sets a digital value (1 = on; 0 = off) | bool | 7e05 | 32261 | Not applicable |
| ModbusMaster.11.Data.SlaveDevice | Slave device to communicate with. | uint8 | 7b1e | 31518 | Not applicable |
| ModbusMaster.11.Data.Status | Transaction status (as for Modbus Master.1 | uint8 | 7ce1 | 31969 | Not applicable |
| ModbusMaster.11.Data.Value | The value to be written to the slave device | float32 | 7c56 | 31830 | 2dp |
| ModbusMaster.12.Data.AlarmStatus | Alarm status (as for Modbus Master.1) | uint8 | 7dc6 | 32198 | Not applicable |
| ModbusMaster.12.Data.BitPosition | Bit position of the bit of interest in a 16 bit data type | uint8 | 7d5a | 32090 | Not applicable |
| ModbusMaster.12.Data.ChanAlarmStatus | Channel alarm status (as for Modbus Master.1) | uint8 | 7de6 | 32230 | Not applicable |
| ModbusMaster.12.Data.DataType | Type of data being read/written (as for Modbus Master.1) | uint8 | 7c11 | 31761 | Not applicable |
| ModbusMaster.12.Data.Descriptor | Description for this data item | string_t | 676e | 26478 | Not applicable |
| ModbusMaster.12.Data.Digital | Digital status (0 = Off, 1 = On) | bool | 7e26 | 32294 | Not applicable |
| ModbusMaster.12.Data.FallBackValue | Fall back value to be writen to the slave device | float32 | 7c94 | 31892 | 2dp |
| ModbusMaster.12.Data.FunctionCode | The modbus function code (as for Modbus Master.1) | uint8 | 7bf3 | 31731 | Not applicable |
| ModbusMaster.12.Data.ModbusAddress | Modbus register address of the data to be read/written | float32 | 7bc2 | 31682 | 0dp |
| ModbusMaster.12.Data.Mode | Auto Manual mode selection (0 = Auto; 1 = Manual) | uint8 | 7da6 | 32166 | Not applicable |
| ModbusMaster.12.Data.Number | Used for multiple instance parameters | uint8 | 7d1e | 32030 | Not applicable |
| ModbusMaster.12.Data.ParameterList | Parameter list for a specific slave device | uint8 | 7d00 | 32000 | Not applicable |
| ModbusMaster.12.Data.Priority | Read/Write frequency (as for Modbus Master.1) | uint8 | 7c2f | 31791 | Not applicable |
| ModbusMaster.12.Data.PV | Process value recieved from slave device | float32 | 7b48 | 31560 | 2dp |
| ModbusMaster.12.Data.Scaling | Scaling in decimal places for non floating point data types | uint8 | 7d3c | 32060 | Not applicable |
| ModbusMaster.12.Data.Send | 1 = send the write value to the slave | bool | 7cc4 | 31940 | Not applicable |
| ModbusMaster.12.Data.Set | Sets a digital value (1 = on; 0 = off) | bool | 7e06 | 32262 | Not applicable |
| ModbusMaster.12.Data.SlaveDevice | Slave device to communicate with. | uint8 | 7b1f | 31519 | Not applicable |
| ModbusMaster.12.Data.Status | Transaction status (as for Modbus Master.1 | uint8 | 7ce2 | 31970 | Not applicable |
| ModbusMaster.12.Data.Value | The value to be written to the slave device | float32 | 7c58 | 31832 | 2dp |
| ModbusMaster.13.Data.AlarmStatus | Alarm status (as for Modbus Master.1) | uint8 | 7dc7 | 32199 | Not applicable |
| ModbusMaster.13.Data.BitPosition | Bit position of the bit of interest in a 16 bit data type | uint8 | 7d5b | 32091 | Not applicable |
| ModbusMaster.13.Data.ChanAlarmStatus | Channel alarm status (as for Modbus Master.1) | uint8 | 7de7 | 32231 | Not applicable |
| ModbusMaster.13.Data.DataType | Type of data being read/written (as for Modbus Master.1) | uint8 | 7c12 | 31762 | Not applicable |
| ModbusMaster.13.Data.Descriptor | Description for this data item | string_t | 6783 | 26499 | Not applicable |
| ModbusMaster.13.Data.Digital | Digital status (0 = Off, 1 = On) | bool | 7e27 | 32295 | Not applicable |
| ModbusMaster.13.Data.FallBackValue | Fall back value to be writen to the slave device | float32 | 7c96 | 31894 | 2dp |
| ModbusMaster.13.Data.FunctionCode | The modbus function code (as for Modbus Master.1) | uint8 | 7bf4 | 31732 | Not applicable |
| ModbusMaster.13.Data.ModbusAddress | Modbus register address of the data to be read/written | float32 | 7bc4 | 31684 | 0dp |
| ModbusMaster.13.Data.Mode | Auto Manual mode selection (0 = Auto; 1 = Manual) | uint8 | 7da7 | 32167 | Not applicable |
| ModbusMaster.13.Data.Number | Used for multiple instance parameters | uint8 | 7d1f | 32031 | Not applicable |
| ModbusMaster.13.Data.ParameterList | Parameter list for a specific slave device | uint8 | 7d01 | 32001 | Not applicable |
| ModbusMaster.13.Data.Priority | Read/Write frequency (as for Modbus Master.1) | uint8 | 7c30 | 31792 | Not applicable |
| ModbusMaster.13.Data.PV | Process value recieved from slave device | float32 | 7b4a | 31562 | 2dp |
| ModbusMaster.13.Data.Scaling | Scaling in decimal places for non floating point data types | uint8 | 7d3d | 32061 | Not applicable |
| ModbusMaster.13.Data.Send | 1 = send the write value to the slave | bool | 7cc5 | 31941 | Not applicable |
| ModbusMaster.13.Data.Set | SSets a digital value (1 = on; 0 = off) | bool | 7e07 | 32263 | Not applicable |
| ModbusMaster.13.Data.SlaveDevice | Slave device to communicate with. | uint8 | 7b20 | 31520 | Not applicable |
| ModbusMaster.13.Data.Status | Transaction status (as for Modbus Master.1 | uint8 | 7ce3 | 31971 | Not applicable |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| ModbusMaster.13.Data.Value | The value to be written to the slave device | float32 | 7c5a | 31834 | 2dp |
| ModbusMaster.14.Data.AlarmStatus | Alarm status (as for Modbus Master.1) | uint8 | 7dc8 | 32200 | Not applicable |
| ModbusMaster.14.Data.BitPosition | Bit position of the bit of interest in a 16 bit data type | uint8 | 7d5c | 32092 | Not applicable |
| ModbusMaster.14.Data.ChanAlarmStatus | Channel alarm status (as for Modbus Master.1) | uint8 | 7de8 | 32232 | Not applicable |
| ModbusMaster.14.Data.DataType | Type of data being read/written (as for Modbus Master.1) | uint8 | 7c13 | 31763 | Not applicable |
| ModbusMaster.14.Data.Descriptor | Description for this data item | string_t | 6798 | 26520 | Not applicable |
| ModbusMaster.14.Data.Digital | Digital status (0 = Off, 1 = On) | bool | 7e28 | 32296 | Not applicable |
| ModbusMaster.14.Data.FallBackValue | Fall back value to be writen to the slave device | float32 | 7c98 | 31896 | 2dp |
| ModbusMaster.14.Data.FunctionCode | The modbus function code (as for Modbus Master.1) | uint8 | 7bf5 | 31733 | Not applicable |
| ModbusMaster.14.Data.ModbusAddress | Modbus register address of the data to be read/written | float32 | 7bc6 | 31686 | 0dp |
| ModbusMaster.14.Data.Mode | Auto Manual mode selection (0 = Auto; 1 = Manual) | uint8 | 7da8 | 32168 | Not applicable |
| ModbusMaster.14.Data.Number | Used for multiple instance parameters | uint8 | 7d20 | 32032 | Not applicable |
| ModbusMaster.14.Data.ParameterList | Parameter list for a specific slave device | uint8 | 7d02 | 32002 | Not applicable |
| ModbusMaster.14.Data.Priority | Read/Write frequency (as for Modbus Master.1) | uint8 | 7c31 | 31793 | Not applicable |
| ModbusMaster.14.Data.PV | Process value recieved from slave device | float32 | 7b4c | 31564 | 2dp |
| ModbusMaster.14.Data.Scaling | Scaling in decimal places for non floating point data types | uint8 | 7d3e | 32062 | Not applicable |
| ModbusMaster.14.Data.Send | 1 = send the write value to the slave | bool | 7cc6 | 31942 | Not applicable |
| ModbusMaster.14.Data.Set | Sets a digital value (1 = on; 0 = off) | bool | 7e08 | 32264 | Not applicable |
| ModbusMaster.14.Data.SlaveDevice | Slave device to communicate with. | uint8 | 7b21 | 31521 | Not applicable |
| ModbusMaster.14.Data.Status | Transaction status (as for Modbus Master.1 | uint8 | 7ce4 | 31972 | Not applicable |
| ModbusMaster.14.Data.Value | The value to be written to the slave device | float32 | 7c5c | 31836 | 2dp |
| ModbusMaster.15.Data.AlarmStatus | Alarm status (as for Modbus Master.1) | uint8 | 7dc9 | 32201 | Not applicable |
| ModbusMaster.15.Data.BitPosition | Bit position of the bit of interest in a 16 bit data type | uint8 | 7d5d | 32093 | Not applicable |
| ModbusMaster.15.Data.ChanAlarmStatus | Channel alarm status (as for Modbus Master.1) | uint8 | 7de9 | 32233 | Not applicable |
| ModbusMaster.15.Data.DataType | Type of data being read/written (as for Modbus Master.1) | uint8 | 7c14 | 31764 | Not applicable |
| ModbusMaster.15.Data.Descriptor | Description for this data item | string_t | 67ad | 26541 | Not applicable |
| ModbusMaster.15.Data.Digital | Digital status (0 = Off, 1 = On) | bool | 7e29 | 32297 | Not applicable |
| ModbusMaster.15.Data.FallBackValue | Fall back value to be writen to the slave device | float32 | 7c9a | 31898 | 2dp |
| ModbusMaster.15.Data.FunctionCode | The modbus function code (as for Modbus Master.1) | uint8 | 7bf6 | 31734 | Not applicable |
| ModbusMaster.15.Data.ModbusAddress | Modbus register address of the data to be read/written | float32 | 7bc8 | 31688 | 0dp |
| ModbusMaster.15.Data.Mode | Auto Manual mode selection (0 = Auto; 1 = Manual) | uint8 | 7da9 | 32169 | Not applicable |
| ModbusMaster.15.Data.Number | Used for multiple instance parameters | uint8 | 7d21 | 32033 | Not applicable |
| ModbusMaster.15.Data.ParameterList | Parameter list for a specific slave device | uint8 | 7d03 | 32003 | Not applicable |
| ModbusMaster.15.Data.Priority | Read/Write frequency (as for Modbus Master.1) | uint8 | 7c32 | 31794 | Not applicable |
| ModbusMaster.15.Data.PV | Process value recieved from slave device | float32 | 7b4e | 31566 | 2dp |
| ModbusMaster.15.Data.Scaling | Scaling in decimal places for non floating point data types | uint8 | 7d3f | 32063 | Not applicable |
| ModbusMaster.15.Data.Send | 1 = send the write value to the slave | bool | 7cc7 | 31943 | Not applicable |
| ModbusMaster.15.Data.Set | Sets a digital value (1 = on; 0 = off) | bool | 7e09 | 32265 | Not applicable |
| ModbusMaster.15.Data.SlaveDevice | Slave device to communicate with. | uint8 | 7b22 | 31522 | Not applicable |
| ModbusMaster.15.Data.Status | Transaction status (as for Modbus Master.1 | uint8 | 7ce5 | 31973 | Not applicable |
| ModbusMaster.15.Data.Value | The value to be written to the slave device | float32 | 7c5e | 31838 | 2dp |
| ModbusMaster.16.Data.AlarmStatus | Alarm status (as for Modbus Master.1) | uint8 | 7dca | 32202 | Not applicable |
| ModbusMaster.16.Data.BitPosition | Bit position of the bit of interest in a 16 bit data type | uint8 | 7d5e | 32094 | Not applicable |
| ModbusMaster.16.Data.ChanAlarmStatus | Channel alarm status (as for Modbus Master.1) | uint8 | 7dea | 32234 | Not applicable |
| ModbusMaster.16.Data.DataType | Type of data being read/written (as for Modbus Master.1) | uint8 | 7c15 | 31765 | Not applicable |
| ModbusMaster.16.Data.Descriptor | Description for this data item | string_t | 67c2 | 26562 | Not applicable |
| ModbusMaster.16.Data.Digital | Digital status (0 = Off, 1 = On) | bool | 7e2a | 32298 | Not applicable |
| ModbusMaster.16.Data.FallBackValue | Fall back value to be writen to the slave device | float32 | 7c9c | 31900 | 2dp |
| ModbusMaster.16.Data.FunctionCode | The modbus function code (as for Modbus Master.1) | uint8 | 7bf7 | 31735 | Not applicable |
| ModbusMaster.16.Data.ModbusAddress | Modbus register address of the data to be read/written | float32 | 7bca | 31690 | 0dp |
| ModbusMaster.16.Data.Mode | Auto Manual mode selection (0 = Auto; 1 = Manual) | uint8 | 7daa | 32170 | Not applicable |
| ModbusMaster.16.Data.Number | Used for multiple instance parameters | uint8 | 7d22 | 32034 | Not applicable |
| ModbusMaster.16.Data.ParameterList | Parameter list for a specific slave device | uint8 | 7d04 | 32004 | Not applicable |
| ModbusMaster.16.Data.Priority | Read/Write frequency (as for Modbus Master.1) | uint8 | 7c33 | 31795 | Not applicable |
| ModbusMaster.16.Data.PV | Process value recieved from slave device | float32 | 7b50 | 31568 | 2dp |
| ModbusMaster.16.Data.Scaling | Scaling in decimal places for non floating point data types | uint8 | 7d40 | 32064 | Not applicable |
| ModbusMaster.16.Data.Send | 1 = send the write value to the slave | bool | 7cc8 | 31944 | Not applicable |
| ModbusMaster.16.Data.Set | Sets a digital value (1 = on; 0 = off) | bool | 7e0a | 32266 | Not applicable |
| ModbusMaster.16.Data.SlaveDevice | Slave device to communicate with. | uint8 | 7b23 | 31523 | Not applicable |
| ModbusMaster.16.Data.Status | Transaction status (as for Modbus Master.1 | uint8 | 7ce6 | 31974 | Not applicable |
| ModbusMaster.16.Data.Value | The value to be written to the slave device | float32 | 7c60 | 31840 | 2dp |
| ModbusMaster.17.Data.AlarmStatus | Alarm status (as for Modbus Master.1) | uint8 | 7dcb | 32203 | Not applicable |
| ModbusMaster.17.Data.BitPosition | Bit position of the bit of interest in a 16 bit data type | uint8 | 7d5f | 32095 | Not applicable |
| ModbusMaster.17.Data.ChanAlarmStatus | Channel alarm status (as for Modbus Master.1) | uint8 | 7deb | 32235 | Not applicable |
| ModbusMaster.17.Data.DataType | Type of data being read/written (as for Modbus Master.1) | uint8 | 7c16 | 31766 | Not applicable |
| ModbusMaster.17.Data.Descriptor | Description for this data item | string_t | 67d7 | 26583 | Not applicable |
| ModbusMaster.17.Data.Digital | Digital status (0 = Off, 1 = On) | bool | 7e2b | 32299 | Not applicable |
| ModbusMaster.17.Data.FallBackValue | Fall back value to be writen to the slave device | float32 | 7c9e | 31902 | 2dp |
| ModbusMaster.17.Data.FunctionCode | The modbus function code (as for Modbus Master.1) | uint8 | 7bf8 | 31736 | Not applicable |
| ModbusMaster.17.Data.ModbusAddress | Modbus register address of the data to be read/written | float32 | 7bcc | 31692 | 0dp |
| ModbusMaster.17.Data.Mode | Auto Manual mode selection (0 = Auto; 1 = Manual) | uint8 | 7dab | 32171 | Not applicable |
| ModbusMaster.17.Data.Number | Used for multiple instance parameters | uint8 | 7d23 | 32035 | Not applicable |
| ModbusMaster.17.Data.ParameterList | Parameter list for a specific slave device | uint8 | 7d05 | 32005 | Not applicable |
| ModbusMaster.17.Data.Priority | Read/Write frequency (as for Modbus Master.1) | uint8 | 7c34 | 31796 | Not applicable |
| ModbusMaster.17.Data.PV | Process value recieved from slave device | float32 | 7b52 | 31570 | 2dp |
| ModbusMaster.17.Data.Scaling | Scaling in decimal places for non floating point data types | uint8 | 7d41 | 32065 | Not applicable |
| ModbusMaster.17.Data.Send | 1 = send the write value to the slave | bool | 7cc9 | 31945 | Not applicable |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| ModbusMaster.17.Data.Set | Sets a digital value (1 = on; 0 = off) | bool | 7e0b | 32267 | Not applicable |
| ModbusMaster.17.Data.SlaveDevice | Slave device to communicate with. | uint8 | 7b24 | 31524 | Not applicable |
| ModbusMaster.17.Data.Status | Transaction status (as for Modbus Master.1 | uint8 | 7ce7 | 31975 | Not applicable |
| ModbusMaster.17.Data.Value | The value to be written to the slave device | float32 | 7c62 | 31842 | 2dp |
| | | | | | |
| ModbusMaster.18.Data.AlarmStatus | Alarm status (as for Modbus Master.1) | uint8 | 7dcc | 32204 | Not applicable |
| ModbusMaster.18.Data.BitPosition | Bit position of the bit of interest in a 16 bit data type | uint8 | 7d60 | 32096 | Not applicable |
| ModbusMaster.18.Data.ChanAlarmStatus | Channel alarm status (as for Modbus Master.1) | uint8 | 7dec | 32236 | Not applicable |
| ModbusMaster.18.Data.DataType | Type of data being read/written (as for Modbus Master.1) | uint8 | 7c17 | 31767 | Not applicable |
| ModbusMaster.18.Data.Descriptor | Description for this data item | string_t | 67ec | 26604 | Not applicable |
| ModbusMaster.18.Data.Digital | Digital status (0 = Off, 1 = On) | bool | 7e2c | 32300 | Not applicable |
| ModbusMaster.18.Data.FallBackValue | Fall back value to be writen to the slave device | float32 | 7ca0 | 31904 | 2dp |
| ModbusMaster.18.Data.FunctionCode | The modbus function code (as for Modbus Master.1) | uint8 | 7bf9 | 31737 | Not applicable |
| ModbusMaster.18.Data.ModbusAddress | Modbus register address of the data to be read/written | float32 | 7bce | 31694 | 0dp |
| ModbusMaster.18.Data.Mode | Auto Manual mode selection (0 = Auto; 1 = Manual) | uint8 | 7dac | 32172 | Not applicable |
| ModbusMaster.18.Data.Number | Used for multiple instance parameters | uint8 | 7d24 | 32036 | Not applicable |
| ModbusMaster.18.Data.ParameterList | Parameter list for a specific slave device | uint8 | 7d06 | 32006 | Not applicable |
| ModbusMaster.18.Data.Priority | FRead/Write frequency (as for Modbus Master.1) | uint8 | 7c35 | 31797 | Not applicable |
| ModbusMaster.18.Data.PV | Process value recieved from slave device | float32 | 7b54 | 31572 | 2dp |
| ModbusMaster.18.Data.Scaling | Scaling in decimal places for non floating point data types | uint8 | 7d42 | 32066 | Not applicable |
| ModbusMaster.18.Data.Send | 1 = send the write value to the slave device | bool | 7cca | 31946 | Not applicable |
| ModbusMaster.18.Data.Set | Sets a digital value (1 = on; 0 = off) | bool | 7e0c | 32268 | Not applicable |
| ModbusMaster.18.Data.SlaveDevice | Slave device to communicate with. | uint8 | 7b25 | 31525 | Not applicable |
| ModbusMaster.18.Data.Status | Transaction status (as for Modbus Master.1 | uint8 | 7ce8 | 31976 | Not applicable |
| ModbusMaster.18.Data.Value | The value to be written to the slave device | float32 | 7c64 | 31844 | 2dp |
| | | | | | |
| ModbusMaster.19.Data.AlarmStatus | Alarm status (as for Modbus Master.1) | uint8 | 7dcd | 32205 | Not applicable |
| ModbusMaster.19.Data.BitPosition | Bit position of the bit of interest in a 16 bit data type | uint8 | 7d61 | 32097 | Not applicable |
| ModbusMaster.19.Data.ChanAlarmStatus | Channel alarm status (as for Modbus Master.1) | uint8 | 7ded | 32237 | Not applicable |
| ModbusMaster.19.Data.DataType | Type of data being read/written (as for Modbus Master.1) | uint8 | 7c18 | 31768 | Not applicable |
| ModbusMaster.19.Data.Descriptor | Description for this data item | string_t | 6801 | 26625 | Not applicable |
| ModbusMaster.19.Data.Digital | Digital status (0 = Off, 1 = On) | bool | 7e2d | 32301 | Not applicable |
| ModbusMaster.19.Data.FallBackValue | Fall back value to be writen to the slave device | float32 | 7ca2 | 31906 | 2dp |
| ModbusMaster.19.Data.FunctionCode | The modbus function code (as for Modbus Master.1) | uint8 | 7bfa | 31738 | Not applicable |
| ModbusMaster.19.Data.ModbusAddress | Modbus register address of the data to be read/written | float32 | 7bd0 | 31696 | 0dp |
| ModbusMaster.19.Data.Mode | Auto Manual mode selection (0 = Auto; 1 = Manual) | uint8 | 7dad | 32173 | Not applicable |
| ModbusMaster.19.Data.Number | Used for multiple instance parameters | uint8 | 7d25 | 32037 | Not applicable |
| ModbusMaster.19.Data.ParameterList | Parameter list for a specific slave device | uint8 | 7d07 | 32007 | Not applicable |
| ModbusMaster.19.Data.Priority | Read/Write frequency (as for Modbus Master.1) | uint8 | 7c36 | 31798 | Not applicable |
| ModbusMaster.19.Data.PV | Process value recieved from slave device | float32 | 7b56 | 31574 | 2dp |
| ModbusMaster.19.Data.Scaling | Scaling in decimal places for non floating point data types | uint8 | 7d43 | 32067 | Not applicable |
| ModbusMaster.19.Data.Send | 1 = send the write value to the slave | bool | 7ccb | 31947 | Not applicable |
| ModbusMaster.19.Data.Set | Sets a digital value (1 = on; 0 = off) | bool | 7e0d | 32269 | Not applicable |
| ModbusMaster.19.Data.SlaveDevice | Slave device to communicate with. | uint8 | 7b26 | 31526 | Not applicable |
| ModbusMaster.19.Data.Status | Transaction status (as for Modbus Master.1 | uint8 | 7ce9 | 31977 | Not applicable |
| ModbusMaster.19.Data.Value | The value to be written to the slave device | float32 | 7c66 | 31846 | 2dp |
| | | | | | |
| ModbusMaster.20.Data.AlarmStatus | Alarm status (as for Modbus Master.1) | uint8 | 7dce | 32206 | Not applicable |
| ModbusMaster.20.Data.BitPosition | Bit position of the bit of interest in a 16 bit data type | uint8 | 7d62 | 32098 | Not applicable |
| ModbusMaster.20.Data.ChanAlarmStatus | Channel alarm status (as for Modbus Master.1) | uint8 | 7dee | 32238 | Not applicable |
| ModbusMaster.20.Data.DataType | Type of data being read/written (as for Modbus Master.1) | uint8 | 7c19 | 31769 | Not applicable |
| ModbusMaster.20.Data.Descriptor | Description for this data item | string_t | 6816 | 26646 | Not applicable |
| ModbusMaster.20.Data.Digital | Digital status (0 = Off, 1 = On) | bool | 7e2e | 32302 | Not applicable |
| ModbusMaster.20.Data.FallBackValue | Fall back value to be writen to the slave device | float32 | 7ca4 | 31908 | 2dp |
| ModbusMaster.20.Data.FunctionCode | The modbus function code (as for Modbus Master.1) | uint8 | 7bfb | 31739 | Not applicable |
| ModbusMaster.20.Data.ModbusAddress | Modbus register address of the data to be read/written | float32 | 7bd2 | 31698 | 0dp |
| ModbusMaster.20.Data.Mode | Auto Manual mode selection (0 = Auto; 1 = Manual) | uint8 | 7dae | 32174 | Not applicable |
| ModbusMaster.20.Data.Number | Used for multiple instance parameters | uint8 | 7d26 | 32038 | Not applicable |
| ModbusMaster.20.Data.ParameterList | Parameter list for a specific slave device | uint8 | 7d08 | 32008 | Not applicable |
| ModbusMaster.20.Data.Priority | Read/Write frequency (as for Modbus Master.1) | uint8 | 7c37 | 31799 | Not applicable |
| ModbusMaster.20.Data.PV | Process value recieved from slave device | float32 | 7b58 | 31576 | 2dp |
| ModbusMaster.20.Data.Scaling | Scaling in decimal places for non floating point data types | uint8 | 7d44 | 32068 | Not applicable |
| ModbusMaster.20.Data.Send | 1 = send the write value to the slave | bool | 7ccc | 31948 | Not applicable |
| ModbusMaster.20.Data.Set | SSets a digital value (1 = on; 0 = off) | bool | 7e0e | 32270 | Not applicable |
| ModbusMaster.20.Data.SlaveDevice | Slave device to communicate with. | uint8 | 7b27 | 31527 | Not applicable |
| ModbusMaster.20.Data.Status | Transaction status (as for Modbus Master.1 | uint8 | 7cea | 31978 | Not applicable |
| ModbusMaster.20.Data.Value | The value to be written to the slave device | float32 | 7c68 | 31848 | 2dp |
| | | | | | |
| ModbusMaster.21.Data.AlarmStatus | Alarm status (as for Modbus Master.1) | uint8 | 7dcf | 32207 | Not applicable |
| ModbusMaster.21.Data.BitPosition | Bit position of the bit of interest in a 16 bit data type | uint8 | 7d63 | 32099 | Not applicable |
| ModbusMaster.21.Data.ChanAlarmStatus | Channel alarm status (as for Modbus Master.1) | uint8 | 7def | 32239 | Not applicable |
| ModbusMaster.21.Data.DataType | Type of data being read/written (as for Modbus Master.1) | uint8 | 7c1a | 31770 | Not applicable |
| ModbusMaster.21.Data.Descriptor | Description for this data item | string_t | 682b | 26667 | Not applicable |
| ModbusMaster.21.Data.Digital | Digital status (0 = Off, 1 = On) | bool | 7e2f | 32303 | Not applicable |
| ModbusMaster.21.Data.FallBackValue | Fall back value to be writen to the slave device | float32 | 7ca6 | 31910 | 2dp |
| ModbusMaster.21.Data.FunctionCode | The modbus function code (as for Modbus Master.1) | uint8 | 7bfc | 31740 | Not applicable |
| ModbusMaster.21.Data.ModbusAddress | Modbus register address of the data to be read/written | float32 | 7bd4 | 31700 | 0dp |
| ModbusMaster.21.Data.Mode | Auto Manual mode selection (0 = Auto; 1 = Manual) | uint8 | 7daf | 32175 | Not applicable |
| ModbusMaster.21.Data.Number | Used for multiple instance parameters | uint8 | 7d27 | 32039 | Not applicable |
| ModbusMaster.21.Data.ParameterList | Parameter list for a specific slave device | uint8 | 7d09 | 32009 | Not applicable |
| ModbusMaster.21.Data.Priority | Read/Write frequency (as for Modbus Master.1) | uint8 | 7c38 | 31800 | Not applicable |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| ModbusMaster.21.Data.PV | Process value recieved from slave device | float32 | 7b5a | 31578 | 2dp |
| ModbusMaster.21.Data.Scaling | Scaling in decimal places for non floating point data types | uint8 | 7d45 | 32069 | Not applicable |
| ModbusMaster.21.Data.Send | 1 = send the write value to the slave | bool | 7ccd | 31949 | Not applicable |
| ModbusMaster.21.Data.Set | Sets a digital value (1 = on; 0 = off) | bool | 7e0f | 32271 | Not applicable |
| ModbusMaster.21.Data.SlaveDevice | Slave device to communicate with. | uint8 | 7b28 | 31528 | Not applicable |
| ModbusMaster.21.Data.Status | Transaction status (as for Modbus Master.1 | uint8 | 7ceb | 31979 | Not applicable |
| ModbusMaster.21.Data.Value | The value to be written to the slave device | float32 | 7c6a | 31850 | 2dp |
| ModbusMaster.22.Data.AlarmStatus | Alarm status (as for Modbus Master.1) | uint8 | 7dd0 | 32208 | Not applicable |
| ModbusMaster.22.Data.BitPosition | Bit position of the bit of interest in a 16 bit data type | uint8 | 7d64 | 32100 | Not applicable |
| ModbusMaster.22.Data.ChanAlarmStatus | Channel alarm status (as for Modbus Master.1) | uint8 | 7df0 | 32240 | Not applicable |
| ModbusMaster.22.Data.DataType | Type of data being read/written (as for Modbus Master.1) | uint8 | 7c1b | 31771 | Not applicable |
| ModbusMaster.22.Data.Descriptor | Description for this data item | string_t | 6840 | 26688 | Not applicable |
| ModbusMaster.22.Data.Digital | Digital status (0 = Off, 1 = On) | bool | 7e30 | 32304 | Not applicable |
| ModbusMaster.22.Data.FallBackValue | Fall back value to be writen to the slave device | float32 | 7ca8 | 31912 | 2dp |
| ModbusMaster.22.Data.FunctionCode | The modbus function code (as for Modbus Master.1) | uint8 | 7bfd | 31741 | Not applicable |
| ModbusMaster.22.Data.ModbusAddress | Modbus register address of the data to be read/written | float32 | 7bd6 | 31702 | 0dp |
| ModbusMaster.22.Data.Mode | Auto Manual mode selection (0 = Auto; 1 = Manual) | uint8 | 7db0 | 32176 | Not applicable |
| ModbusMaster.22.Data.Number | Used for multiple instance parameters | uint8 | 7d28 | 32040 | Not applicable |
| ModbusMaster.22.Data.ParameterList | Parameter list for a specific slave device | uint8 | 7d0a | 32010 | Not applicable |
| ModbusMaster.22.Data.Priority | Read/Write frequency (as for Modbus Master.1) | uint8 | 7c39 | 31801 | Not applicable |
| ModbusMaster.22.Data.PV | Process value recieved from slave device | float32 | 7b5c | 31580 | 2dp |
| ModbusMaster.22.Data.Scaling | Scaling in decimal places for non floating point data types | uint8 | 7d46 | 32070 | Not applicable |
| ModbusMaster.22.Data.Send | 1 = send the write value to the slave | bool | 7cce | 31950 | Not applicable |
| ModbusMaster.22.Data.Set | Sets a digital value (1 = on; 0 = off) | bool | 7e10 | 32272 | Not applicable |
| ModbusMaster.22.Data.SlaveDevice | Slave device to communicate with. | uint8 | 7b29 | 31529 | Not applicable |
| ModbusMaster.22.Data.Status | Transaction status (as for Modbus Master.1 | uint8 | 7cec | 31980 | Not applicable |
| ModbusMaster.22.Data.Value | The value to be written to the slave device | float32 | 7c6c | 31852 | 2dp |
| ModbusMaster.23.Data.AlarmStatus | Alarm status (as for Modbus Master.1) | uint8 | 7dd1 | 32209 | Not applicable |
| ModbusMaster.23.Data.BitPosition | Bit position of the bit of interest in a 16 bit data type | uint8 | 7d65 | 32101 | Not applicable |
| ModbusMaster.23.Data.ChanAlarmStatus | Channel alarm status (as for Modbus Master.1) | uint8 | 7df1 | 32241 | Not applicable |
| ModbusMaster.23.Data.DataType | Type of data being read/written (as for Modbus Master.1) | uint8 | 7c1c | 31772 | Not applicable |
| ModbusMaster.23.Data.Descriptor | Description for this data item | string_t | 6855 | 26709 | Not applicable |
| ModbusMaster.23.Data.Digital | Digital status (0 = Off, 1 = On) | bool | 7e31 | 32305 | Not applicable |
| ModbusMaster.23.Data.FallBackValue | Fall back value to be writen to the slave device | float32 | 7caa | 31914 | 2dp |
| ModbusMaster.23.Data.FunctionCode | The modbus function code (as for Modbus Master.1) | uint8 | 7bfe | 31742 | Not applicable |
| ModbusMaster.23.Data.ModbusAddress | Modbus register address of the data to be read/written | float32 | 7bd8 | 31704 | 0dp |
| ModbusMaster.23.Data.Mode | Auto Manual mode selection (0 = Auto; 1 = Manual) | uint8 | 7db1 | 32177 | Not applicable |
| ModbusMaster.23.Data.Number | Used for multiple instance parameters | uint8 | 7d29 | 32041 | Not applicable |
| ModbusMaster.23.Data.ParameterList | Parameter list for a specific slave device | uint8 | 7d0b | 32011 | Not applicable |
| ModbusMaster.23.Data.Priority | Read/Write frequency (as for Modbus Master.1) | uint8 | 7c3a | 31802 | Not applicable |
| ModbusMaster.23.Data.PV | Process value recieved from slave device | float32 | 7b5e | 31582 | 2dp |
| ModbusMaster.23.Data.Scaling | Scaling in decimal places for non floating point data types | uint8 | 7d47 | 32071 | Not applicable |
| ModbusMaster.23.Data.Send | 1 = send the write value to the slave | bool | 7ccf | 31951 | Not applicable |
| ModbusMaster.23.Data.Set | Sets a digital value (1 = on; 0 = off) | bool | 7e11 | 32273 | Not applicable |
| ModbusMaster.23.Data.SlaveDevice | Slave device to communicate with. | uint8 | 7b2a | 31530 | Not applicable |
| ModbusMaster.23.Data.Status | Transaction status (as for Modbus Master.1 | uint8 | 7ced | 31981 | Not applicable |
| ModbusMaster.23.Data.Value | The value to be written to the slave device | float32 | 7c6e | 31854 | 2dp |
| ModbusMaster.24.Data.AlarmStatus | Alarm status (as for Modbus Master.1) | uint8 | 7dd2 | 32210 | Not applicable |
| ModbusMaster.24.Data.BitPosition | Bit position of the bit of interest in a 16 bit data type | uint8 | 7d66 | 32102 | Not applicable |
| ModbusMaster.24.Data.ChanAlarmStatus | Channel alarm status (as for Modbus Master.1) | uint8 | 7df2 | 32242 | Not applicable |
| ModbusMaster.24.Data.DataType | Type of data being read/written (as for Modbus Master.1) | uint8 | 7c1d | 31773 | Not applicable |
| ModbusMaster.24.Data.Descriptor | Description for this data item | string_t | 686a | 26730 | Not applicable |
| ModbusMaster.24.Data.Digital | Digital status (0 = Off, 1 = On) | bool | 7e32 | 32306 | Not applicable |
| ModbusMaster.24.Data.FallBackValue | Fall back value to be writen to the slave device | float32 | 7cac | 31916 | 2dp |
| ModbusMaster.24.Data.FunctionCode | The modbus function code (as for Modbus Master.1) | uint8 | 7bff | 31743 | Not applicable |
| ModbusMaster.24.Data.ModbusAddress | Modbus register address of the data to be read/written | float32 | 7bda | 31706 | 0dp |
| ModbusMaster.24.Data.Mode | Auto Manual mode selection (0 = Auto; 1 = Manual) | uint8 | 7db2 | 32178 | Not applicable |
| ModbusMaster.24.Data.Number | Used for multiple instance parameters | uint8 | 7d2a | 32042 | Not applicable |
| ModbusMaster.24.Data.ParameterList | Parameter list for a specific slave device | uint8 | 7d0c | 32012 | Not applicable |
| ModbusMaster.24.Data.Priority | Read/Write frequency (as for Modbus Master.1) | uint8 | 7c3b | 31803 | Not applicable |
| ModbusMaster.24.Data.PV | Process value recieved from slave device | float32 | 7b60 | 31584 | 2dp |
| ModbusMaster.24.Data.Scaling | Scaling in decimal places for non floating point data types | uint8 | 7d48 | 32072 | Not applicable |
| ModbusMaster.24.Data.Send | 1 = send the write value to the slave | bool | 7cd0 | 31952 | Not applicable |
| ModbusMaster.24.Data.Set | Sets a digital value (1 = on; 0 = off) | bool | 7e12 | 32274 | Not applicable |
| ModbusMaster.24.Data.SlaveDevice | Slave device to communicate with. | uint8 | 7b2b | 31531 | Not applicable |
| ModbusMaster.24.Data.Status | Transaction status (as for Modbus Master.1 | uint8 | 7cee | 31982 | Not applicable |
| ModbusMaster.24.Data.Value | The value to be written to the slave device | float32 | 7c70 | 31856 | 2dp |
| ModbusMaster.25.Data.AlarmStatus | Alarm status (as for Modbus Master.1) | uint8 | 7dd3 | 32211 | Not applicable |
| ModbusMaster.25.Data.BitPosition | Bit position of the bit of interest in a 16 bit data type | uint8 | 7d67 | 32103 | Not applicable |
| ModbusMaster.25.Data.ChanAlarmStatus | Channel alarm status (as for Modbus Master.1) | uint8 | 7df3 | 32243 | Not applicable |
| ModbusMaster.25.Data.DataType | Type of data being read/written (as for Modbus Master.1) | uint8 | 7c1e | 31774 | Not applicable |
| ModbusMaster.25.Data.Descriptor | Description for this data item | string_t | 687f | 26751 | Not applicable |
| ModbusMaster.25.Data.Digital | Digital status (0 = Off, 1 = On) | bool | 7e33 | 32307 | Not applicable |
| ModbusMaster.25.Data.FallBackValue | Fall back value to be writen to the slave device | float32 | 7cae | 31918 | 2dp |
| ModbusMaster.25.Data.FunctionCode | The modbus function code (as for Modbus Master.1) | uint8 | 7c00 | 31744 | Not applicable |
| ModbusMaster.25.Data.ModbusAddress | Modbus register address of the data to be read/written | float32 | 7bdc | 31708 | 0dp |
| ModbusMaster.25.Data.Mode | Auto Manual mode selection (0 = Auto; 1 = Manual) | uint8 | 7db3 | 32179 | Not applicable |
| ModbusMaster.25.Data.Number | Used for multiple instance parameters | uint8 | 7d2b | 32043 | Not applicable |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| ModbusMaster.25.Data.ParameterList | Parameter list for a specific slave device | uint8 | 7d0d | 32013 | Not applicable |
| ModbusMaster.25.Data.Priority | Read/Write frequency (as for Modbus Master.1) | uint8 | 7c3c | 31804 | Not applicable |
| ModbusMaster.25.Data.PV | Process value recieved from slave device | float32 | 7b62 | 31586 | 2dp |
| ModbusMaster.25.Data.Scaling | Scaling in decimal places for non floating point data types | uint8 | 7d49 | 32073 | Not applicable |
| ModbusMaster.25.Data.Send | 1 = send the write value to the slave | bool | 7cd1 | 31953 | Not applicable |
| ModbusMaster.25.Data.Set | Sets a digital value (1 = on; 0 = off) | bool | 7e13 | 32275 | Not applicable |
| ModbusMaster.25.Data.SlaveDevice | Slave device to communicate with. | uint8 | 7b2c | 31532 | Not applicable |
| ModbusMaster.25.Data.Status | Transaction status (as for Modbus Master.1 | uint8 | 7cef | 31983 | Not applicable |
| ModbusMaster.25.Data.Value | The value to be written to the slave device | float32 | 7c72 | 31858 | 2dp |
| ModbusMaster.26.Data.AlarmStatus | Alarm status (as for Modbus Master.1) | uint8 | 7dd4 | 32212 | Not applicable |
| ModbusMaster.26.Data.BitPosition | Bit position of the bit of interest in a 16 bit data type | uint8 | 7d68 | 32104 | Not applicable |
| ModbusMaster.26.Data.ChanAlarmStatus | Channel alarm status (as for Modbus Master.1) | uint8 | 7df4 | 32244 | Not applicable |
| ModbusMaster.26.Data.DataType | Type of data being read/written (as for Modbus Master.1) | uint8 | 7c1f | 31775 | Not applicable |
| ModbusMaster.26.Data.Descriptor | Description for this data item | string_t | 6894 | 26772 | Not applicable |
| ModbusMaster.26.Data.Digital | Digital status (0 = Off, 1 = On) | bool | 7e34 | 32308 | Not applicable |
| ModbusMaster.26.Data.FallBackValue | Fall back value to be writen to the slave device | float32 | 7cb0 | 31920 | 2dp |
| ModbusMaster.26.Data.FunctionCode | The modbus function code (as for Modbus Master.1) | uint8 | 7c01 | 31745 | Not applicable |
| ModbusMaster.26.Data.ModbusAddress | Modbus register address of the data to be read/written | float32 | 7bde | 31710 | 0dp |
| ModbusMaster.26.Data.Mode | Auto Manual mode selection (0 = Auto; 1 = Manual) | uint8 | 7db4 | 32180 | Not applicable |
| ModbusMaster.26.Data.Number | Used for multiple instance parameters | uint8 | 7d2c | 32044 | Not applicable |
| ModbusMaster.26.Data.ParameterList | Parameter list for a specific slave device | uint8 | 7d0e | 32014 | Not applicable |
| ModbusMaster.26.Data.Priority | Read/Write frequency (as for Modbus Master.1) | uint8 | 7c3d | 31805 | Not applicable |
| ModbusMaster.26.Data.PV | Process value recieved from slave device | float32 | 7b64 | 31588 | 2dp |
| ModbusMaster.26.Data.Scaling | Scaling in decimal places for non floating point data types | uint8 | 7d4a | 32074 | Not applicable |
| ModbusMaster.26.Data.Send | 1 = send the write value to the slave | bool | 7cd2 | 31954 | Not applicable |
| ModbusMaster.26.Data.Set | Sets a digital value (1 = on; 0 = off) | bool | 7e14 | 32276 | Not applicable |
| ModbusMaster.26.Data.SlaveDevice | Slave device to communicate with. | uint8 | 7b2d | 31533 | Not applicable |
| ModbusMaster.26.Data.Status | Transaction status (as for Modbus Master.1 | uint8 | 7cf0 | 31984 | Not applicable |
| ModbusMaster.26.Data.Value | The value to be written to the slave device | float32 | 7c74 | 31860 | 2dp |
| ModbusMaster.27.Data.AlarmStatus | Alarm status (as for Modbus Master.1) | uint8 | 7dd5 | 32213 | Not applicable |
| ModbusMaster.27.Data.BitPosition | Bit position of the bit of interest in a 16 bit data type | uint8 | 7d69 | 32105 | Not applicable |
| ModbusMaster.27.Data.ChanAlarmStatus | Channel alarm status (as for Modbus Master.1) | uint8 | 7df5 | 32245 | Not applicable |
| ModbusMaster.27.Data.DataType | Type of data being read/written (as for Modbus Master.1) | uint8 | 7c20 | 31776 | Not applicable |
| ModbusMaster.27.Data.Descriptor | Description for this data item | string_t | 68a9 | 26793 | Not applicable |
| ModbusMaster.27.Data.Digital | Digital status (0 = Off, 1 = On) | bool | 7e35 | 32309 | Not applicable |
| ModbusMaster.27.Data.FallBackValue | Fall back value to be writen to the slave device | float32 | 7cb2 | 31922 | 2dp |
| ModbusMaster.27.Data.FunctionCode | The modbus function code (as for Modbus Master.1) | uint8 | 7c02 | 31746 | Not applicable |
| ModbusMaster.27.Data.ModbusAddress | Modbus register address of the data to be read/written | float32 | 7be0 | 31712 | 0dp |
| ModbusMaster.27.Data.Mode | Auto Manual mode selection (0 = Auto; 1 = Manual) | uint8 | 7db5 | 32181 | Not applicable |
| ModbusMaster.27.Data.Number | Used for multiple instance parameters | uint8 | 7d2d | 32045 | Not applicable |
| ModbusMaster.27.Data.ParameterList | Parameter list for a specific slave device | uint8 | 7d0f | 32015 | Not applicable |
| ModbusMaster.27.Data.Priority | Read/Write frequency (as for Modbus Master.1) | uint8 | 7c3e | 31806 | Not applicable |
| ModbusMaster.27.Data.PV | Process value recieved from slave device | float32 | 7b66 | 31590 | 2dp |
| ModbusMaster.27.Data.Scaling | Scaling in decimal places for non floating point data types | uint8 | 7d4b | 32075 | Not applicable |
| ModbusMaster.27.Data.Send | 1 = send the write value to the slave | bool | 7cd3 | 31955 | Not applicable |
| ModbusMaster.27.Data.Set | Sets a digital value (1 = on; 0 = off) | bool | 7e15 | 32277 | Not applicable |
| ModbusMaster.27.Data.SlaveDevice | Slave device to communicate with. | uint8 | 7b2e | 31534 | Not applicable |
| ModbusMaster.27.Data.Status | Transaction status (as for Modbus Master.1 | uint8 | 7cf1 | 31985 | Not applicable |
| ModbusMaster.27.Data.Value | The value to be written to the slave device | float32 | 7c76 | 31862 | 2dp |
| ModbusMaster.28.Data.AlarmStatus | Alarm status (as for Modbus Master.1) | uint8 | 7dd6 | 32214 | Not applicable |
| ModbusMaster.28.Data.BitPosition | Bit position of the bit of interest in a 16 bit data type | uint8 | 7d6a | 32106 | Not applicable |
| ModbusMaster.28.Data.ChanAlarmStatus | Channel alarm status (as for Modbus Master.1) | uint8 | 7df6 | 32246 | Not applicable |
| ModbusMaster.28.Data.DataType | Type of data being read/written (as for Modbus Master.1) | uint8 | 7c21 | 31777 | Not applicable |
| ModbusMaster.28.Data.Descriptor | Description for this data item | string_t | 68be | 26814 | Not applicable |
| ModbusMaster.28.Data.Digital | Digital status (0 = Off, 1 = On) | bool | 7e36 | 32310 | Not applicable |
| ModbusMaster.28.Data.FallBackValue | Fall back value to be writen to the slave device | float32 | 7cb4 | 31924 | 2dp |
| ModbusMaster.28.Data.FunctionCode | The modbus function code (as for Modbus Master.1) | uint8 | 7c03 | 31747 | Not applicable |
| ModbusMaster.28.Data.ModbusAddress | Modbus register address of the data to be read/written | float32 | 7be2 | 31714 | 0dp |
| ModbusMaster.28.Data.Mode | Auto Manual mode selection (0 = Auto; 1 = Manual) | uint8 | 7db6 | 32182 | Not applicable |
| ModbusMaster.28.Data.Number | Used for multiple instance parameters | uint8 | 7d2e | 32046 | Not applicable |
| ModbusMaster.28.Data.ParameterList | Parameter list for a specific slave device | uint8 | 7d10 | 32016 | Not applicable |
| ModbusMaster.28.Data.Priority | Read/Write frequency (as for Modbus Master.1) | uint8 | 7c3f | 31807 | Not applicable |
| ModbusMaster.28.Data.PV | Process value recieved from slave device | float32 | 7b68 | 31592 | 2dp |
| ModbusMaster.28.Data.Scaling | Scaling in decimal places for non floating point data types | uint8 | 7d4c | 32076 | Not applicable |
| ModbusMaster.28.Data.Send | 1 = send the write value to the slave | bool | 7cd4 | 31956 | Not applicable |
| ModbusMaster.28.Data.Set | Sets a digital value (1 = on; 0 = off) | bool | 7e16 | 32278 | Not applicable |
| ModbusMaster.28.Data.SlaveDevice | Slave device to communicate with. | uint8 | 7b2f | 31535 | Not applicable |
| ModbusMaster.28.Data.Status | Transaction status (as for Modbus Master.1 | uint8 | 7cf2 | 31986 | Not applicable |
| ModbusMaster.28.Data.Value | The value to be written to the slave device | float32 | 7c78 | 31864 | 2dp |
| ModbusMaster.29.Data.AlarmStatus | Alarm status (as for Modbus Master.1) | uint8 | 7dd7 | 32215 | Not applicable |
| ModbusMaster.29.Data.BitPosition | Bit position of the bit of interest in a 16 bit data type | uint8 | 7d6b | 32107 | Not applicable |
| ModbusMaster.29.Data.ChanAlarmStatus | Channel alarm status (as for Modbus Master.1) | uint8 | 7df7 | 32247 | Not applicable |
| ModbusMaster.29.Data.DataType | Type of data being read/written (as for Modbus Master.1) | uint8 | 7c22 | 31778 | Not applicable |
| ModbusMaster.29.Data.Descriptor | Description for this data item | string_t | 70ff | 28927 | Not applicable |
| ModbusMaster.29.Data.Digital | Digital status (0 = Off, 1 = On) | bool | 7e37 | 32311 | Not applicable |
| ModbusMaster.29.Data.FallBackValue | Fall back value to be writen to the slave device | float32 | 7cb6 | 31926 | 2dp |
| ModbusMaster.29.Data.FunctionCode | The modbus function code (as for Modbus Master.1) | uint8 | 7c04 | 31748 | Not applicable |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| ModbusMaster.29.Data.ModbusAddress | Modbus register address of the data to be read/written | float32 | 7be4 | 31716 | 0dp |
| ModbusMaster.29.Data.Mode | Auto Manual mode selection (0 = Auto; 1 = Manual) | uint8 | 7db7 | 32183 | Not applicable |
| ModbusMaster.29.Data.Number | Used for multiple instance parameters | uint8 | 7d2f | 32047 | Not applicable |
| ModbusMaster.29.Data.ParameterList | Parameter list for a specific slave device | uint8 | 7d11 | 32017 | Not applicable |
| ModbusMaster.29.Data.Priority | Read/Write frequency (as for Modbus Master.1) | uint8 | 7c40 | 31808 | Not applicable |
| ModbusMaster.29.Data.PV | Process value recieved from slave device | float32 | 7b6a | 31594 | 2dp |
| ModbusMaster.29.Data.Scaling | Scaling in decimal places for non floating point data types | uint8 | 7d4d | 32077 | Not applicable |
| ModbusMaster.29.Data.Send | 1 = send the write value to the slave | bool | 7cd5 | 31957 | Not applicable |
| ModbusMaster.29.Data.Set | Sets a digital value (1 = on; 0 = off) | bool | 7e17 | 32279 | Not applicable |
| ModbusMaster.29.Data.SlaveDevice | Slave device to communicate with. | uint8 | 7b30 | 31536 | Not applicable |
| ModbusMaster.29.Data.Status | Transaction status (as for Modbus Master.1 | uint8 | 7cf3 | 31987 | Not applicable |
| ModbusMaster.29.Data.Value | The value to be written to the slave device | float32 | 7c7a | 31866 | 2dp |
| | | | | | |
| ModbusMaster.30.Data.AlarmStatus | Alarm status (as for Modbus Master.1) | uint8 | 7dd8 | 32216 | Not applicable |
| ModbusMaster.30.Data.BitPosition | Bit position of the bit of interest in a 16 bit data type | uint8 | 7d6c | 32108 | Not applicable |
| ModbusMaster.30.Data.ChanAlarmStatus | Channel alarm status (as for Modbus Master.1) | uint8 | 7df8 | 32248 | Not applicable |
| ModbusMaster.30.Data.DataType | Type of data being read/written (as for Modbus Master.1) | uint8 | 7c23 | 31779 | Not applicable |
| ModbusMaster.30.Data.Descriptor | Description for this data item | string_t | 7114 | 28948 | Not applicable |
| ModbusMaster.30.Data.Digital | Digital status (0 = Off, 1 = On) | bool | 7e38 | 32312 | Not applicable |
| ModbusMaster.30.Data.FallBackValue | Fall back value to be writen to the slave device | float32 | 7cb8 | 31928 | 2dp |
| ModbusMaster.30.Data.FunctionCode | The modbus function code (as for Modbus Master.1) | uint8 | 7c05 | 31749 | Not applicable |
| ModbusMaster.30.Data.ModbusAddress | Modbus register address of the data to be read/written | float32 | 7be6 | 31718 | 0dp |
| ModbusMaster.30.Data.Mode | Auto Manual mode selection (0 = Auto; 1 = Manual) | uint8 | 7db8 | 32184 | Not applicable |
| ModbusMaster.30.Data.Number | Used for multiple instance parameters | uint8 | 7d30 | 32048 | Not applicable |
| ModbusMaster.30.Data.ParameterList | Parameter list for a specific slave device | uint8 | 7d12 | 32018 | Not applicable |
| ModbusMaster.30.Data.Priority | Read/Write frequency (as for Modbus Master.1) | uint8 | 7c41 | 31809 | Not applicable |
| ModbusMaster.30.Data.PV | Process value recieved from slave device | float32 | 7b6c | 31596 | 2dp |
| ModbusMaster.30.Data.Scaling | Scaling in decimal places for non floating point data types | uint8 | 7d4e | 32078 | Not applicable |
| ModbusMaster.30.Data.Send | 1 = send the write value to the slave | bool | 7cd6 | 31958 | Not applicable |
| ModbusMaster.30.Data.Set | Sets a digital value (1 = on; 0 = off) | bool | 7e18 | 32280 | Not applicable |
| ModbusMaster.30.Data.SlaveDevice | Slave device to communicate with. | uint8 | 7b31 | 31537 | Not applicable |
| ModbusMaster.30.Data.Status | Transaction status (as for Modbus Master.1 | uint8 | 7cf4 | 31988 | Not applicable |
| ModbusMaster.30.Data.Value | The value to be written to the slave device | float32 | 7c7c | 31868 | 2dp |
| | | | | | |
| ModbusMaster.Slave1.Data.AlarmStatus | Alarm status (0 = none; 1 = one or more alarms active) | uint8 | 7db9 | 32185 | Not applicable |
| ModbusMaster.Slave1.Data.BitPosition | Bit position of the bit of interest in a 16 bit data type | uint8 | 7d95 | 32149 | Not applicable |
| ModbusMaster.Slave1.Data.ChanAlarmStatus | Channel alarm status<br>0 = Off 1 = Active 2 = Safe Nak'd 3 = Active Nack'd | uint8 | 7dd9 | 32217 | Not applicable |
| ModbusMaster.Slave1.Data.DataType | Data type of the data being read/written<br>0 = Real 1 = DINT 2 = INT 3 = Byte<br>4 = UDINT 5 = UINT 6 = UBYTE 8 = Real (Swap)<br>9 = DINT (Swap) 10 = UDINT (Swap) 11 = BIT | uint8 | 7d7f | 32127 | Not applicable |
| ModbusMaster.Slave1.Data.Descriptor | Description for this data item | string_t | 665d | 26205 | Not applicable |
| ModbusMaster.Slave1.Data.Digital | Digital status (0 = Off; 1 = On) | bool | 7e19 | 32281 | Not applicable |
| ModbusMaster.Slave1.Data.FallBackValue | Fall back value to be writen to the slave device | float32 | 7d87 | 32135 | 2dp |
| ModbusMaster.Slave1.Data.FunctionCode | The modbus function code<br>1 = Read coil 2 = Read discrete 3 = Read holding<br>4 = Read input 5 = Write coil 6 = Write single<br>16 = Write multiple | uint8 | 7d7d | 32125 | Not applicable |
| ModbusMaster.Slave1.Data.ModbusAddress | Modbus register address of the data to be read/written | float32 | 7d79 | 32121 | 0dp |
| ModbusMaster.Slave1.Data.Mode | Auto Manual mode selection (0 = Auto; 1 = Manual) | uint8 | 7d99 | 32153 | Not applicable |
| ModbusMaster.Slave1.Data.Number | Used for multiple instance parameters | uint8 | 7d91 | 32145 | Not applicable |
| ModbusMaster.Slave1.Data.ParameterList | Parameter list for a specific slave device | uint8 | 7d8f | 32143 | Not applicable |
| ModbusMaster.Slave1.Data.Priority | Frequency at which the data is read/written<br>0 = High 1 = Medium 2 = Low 3 = Acyclic | uint8 | 7d81 | 32129 | Not applicable |
| ModbusMaster.Slave1.Data.PV | Process value recieved from slave device | float32 | 7d73 | 32115 | 2dp |
| ModbusMaster.Slave1.Data.Scaling | Scaling in decimal places for non floating point data types | uint8 | 7d93 | 32147 | Not applicable |
| ModbusMaster.Slave1.Data.Send | 1 = send the write value to the slave | bool | 7d8b | 32139 | Not applicable |
| ModbusMaster.Slave1.Data.Set | Sets a digital value to on (1) or off (0) | bool | 7df9 | 32249 | Not applicable |
| ModbusMaster.Slave1.Data.SlaveDevice | Slave device to communicate with. | uint8 | 7d71 | 32113 | Not applicable |
| ModbusMaster.Slave1.Data.Status | Transaction status<br>0 = Success 1 = Illegal function 2 = Ilegal address<br>3 = Illegal value 6 = Slave busy 8 = Parity error<br>9 = Bad sub 10 = Bad gateway 11 = No response<br>12 = Idle 13 = Pending 14 = Timeout<br>15 = Unknown host 16 = Connect fail 17 = No sockets<br>18 = Loopback fail 19 = Login fail 20 = Unknown error<br>22 = Write fail 23 = Master reject | uint8 | 7d8d | 32141 | Not applicable |
| ModbusMaster.Slave1.Data.Value | The value to be written to the slave device | float32 | 7d83 | 32131 | 2dp |
| ModbusMaster.Slave1.Main.CommsFailure | 1 = a device communications failure | bool | 7d97 | 32151 | Not applicable |
| ModbusMaster.Slave1.Main.Descriptor | Device descriptor | string_t | 6633 | 26163 | Not applicable |
| ModbusMaster.Slave1.Main.HighPriority | High priority rate<br>0 = 125ms 1 = 250ms 2 = 500 ms 3 = 1 sec<br>4 = 2 secs 5 = 5 secs 6 = 10 secs 7 = 20 secs<br>8 = 30 secs 9 = 1 min 10 = 2 mins 11 = 5 mins<br>12 = 10 mins 13 = 20 mins 14 = 30 mins 15 = 1 hr | uint8 | 7b0c | 31500 | Not applicable |
| ModbusMaster.Slave1.Main.IPAddress | Internet Protocol (IP) address for a slave device | string_t | 68d3 | 26835 | Not applicable |
| ModbusMaster.Slave1.Main.LowPriority | Low priority rate (as 'high priority' above) | uint8 | 7b10 | 31504 | Not applicable |
| ModbusMaster.Slave1.Main.MaxBlockSize | Maximum amount of data in a single transaction | uint8 | 7b0a | 31498 | Not applicable |
| ModbusMaster.Slave1.Main.MediumPriority | Medium priority rate (as 'high priority' above) | uint8 | 7b0e | 31502 | Not applicable |
| ModbusMaster.Slave1.Main.Online | Enables communications (0 = offline; 1 = online) | bool | 7b00 | 31488 | Not applicable |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| ModbusMaster.Slave1.Main.Profile | A profile that defines the device type<br>0 = 3rd party 1 = Mini8 2 = 3xxx 3 = 35xx<br>4 = 2xxx 5 = 2500 6 = 5000 7 = 6000<br>8 = nanodac 9 = EPower | uint8 | 7b12 | 31506 | Not applicable |
| ModbusMaster.Slave1.Main.Retries | Transaction retries | uint8 | 7b04 | 31492 | Not applicable |
| ModbusMaster.Slave1.Main.SearchDevice | Initiates a slave search (0 = No; 1 = Yes) | bool | 7d6d | 32109 | Not applicable |
| ModbusMaster.Slave1.Main.SearchResult | Current search status<br>0 = Searching 1 = Available 2 = Unavailable<br>3 = Unreachable 4 = Aborted | uint8 | 7d6f | 32111 | Not applicable |
| ModbusMaster.Slave1.Main.Timeout | Time in milliseconds the master will wait for a response | float32 | 7b06 | 31494 | 0dp |
| ModbusMaster.Slave1.Main.UnitId | Unit id for a slave device | uint8 | 7b02 | 31490 | Not applicable |
| ModbusMaster.Slave2.Data.AlarmStatus | Alarm status (0 = none; 1 = one or more alarms active) | uint8 | 7dba | 32186 | Not applicable |
| ModbusMaster.Slave2.Data.BitPosition | Bit position of the bit of interest in a 16 bit data type | uint8 | 7d96 | 32150 | Not applicable |
| ModbusMaster.Slave2.Data.ChanAlarmStatus | Channel alarm status (as Slave1.Data) | uint8 | 7dda | 32218 | Not applicable |
| ModbusMaster.Slave2.Data.DataType | Data type of the data being read/written (as Slave1.Data) | uint8 | 7d80 | 32128 | Not applicable |
| ModbusMaster.Slave2.Data.Descriptor | Description for this data item | string_t | 6672 | 26226 | Not applicable |
| ModbusMaster.Slave2.Data.Digital | Digital status (0 = Off; 1 = On) | bool | 7e1a | 32282 | Not applicable |
| ModbusMaster.Slave2.Data.FallBackValue | Fall back value to be writen to the slave device | float32 | 7d89 | 32137 | 2dp |
| ModbusMaster.Slave2.Data.FunctionCode | The modbus function code (as Slave1.Data) | uint8 | 7d7e | 32126 | Not applicable |
| ModbusMaster.Slave2.Data.ModbusAddress | Modbus register address of the data to be read/written | float32 | 7d7b | 32123 | 0dp |
| ModbusMaster.Slave2.Data.Mode | Auto Manual mode selectionn (0 = Auto; 1 = Manual) | uint8 | 7d9a | 32154 | Not applicable |
| ModbusMaster.Slave2.Data.Number | Used for multiple instance parameters | uint8 | 7d92 | 32146 | Not applicable |
| ModbusMaster.Slave2.Data.ParameterList | Parameter list for a specific slave device | uint8 | 7d90 | 32144 | Not applicable |
| ModbusMaster.Slave2.Data.Priority | Frequency at which the data is read/written (as Slave1.Data) | uint8 | 7d82 | 32130 | Not applicable |
| ModbusMaster.Slave2.Data.PV | Process value recieved from slave device | float32 | 7d75 | 32117 | 2dp |
| ModbusMaster.Slave2.Data.Scaling | Scaling in decimal places for non floating point data types | uint8 | 7d94 | 32148 | Not applicable |
| ModbusMaster.Slave2.Data.Send | 1 = send the write value to the slave | bool | 7d8c | 32140 | Not applicable |
| ModbusMaster.Slave2.Data.Set | SSets a digital value to on (1) or off (0) | bool | 7dfa | 32250 | Not applicable |
| ModbusMaster.Slave2.Data.SlaveDevice | Slave device to communicate with. | uint8 | 7d72 | 32114 | Not applicable |
| ModbusMaster.Slave2.Data.Status | Transaction status (as for Slave 1) | uint8 | 7d8e | 32142 | Not applicable |
| ModbusMaster.Slave2.Data.Value | The value to be written to the slave device | float32 | 7d85 | 32133 | 2dp |
| ModbusMaster.Slave2.Main.CommsFailure | 1 = a device communications failure | bool | 7d98 | 32152 | Not applicable |
| ModbusMaster.Slave2.Main.Descriptor | Device descriptor | string_t | 6648 | 26184 | Not applicable |
| ModbusMaster.Slave2.Main.HighPriority | High priority rate (as for Slave 1) | uint8 | 7b0d | 31501 | Not applicable |
| ModbusMaster.Slave2.Main.IPAddress | Internet Protocol (IP) address for a slave device | string_t | 68e5 | 26853 | Not applicable |
| ModbusMaster.Slave2.Main.LowPriority | Low priority rate (as for Slave 1) | uint8 | 7b11 | 31505 | Not applicable |
| ModbusMaster.Slave2.Main.MaxBlockSize | Maximum amount of data in a single transaction | uint8 | 7b0b | 31499 | Not applicable |
| ModbusMaster.Slave2.Main.MediumPriority | Medium priority rate (as for Slave 1) | uint8 | 7b0f | 31503 | Not applicable |
| ModbusMaster.Slave2.Main.Online | Enables communications (0 = offline; 1 = online) | bool | 7b01 | 31489 | Not applicable |
| ModbusMaster.Slave2.Main.Profile | A profile that defines the device type (as Slave1.Data) | uint8 | 7b13 | 31507 | Not applicable |
| ModbusMaster.Slave2.Main.Retries | Transaction retries | uint8 | 7b05 | 31493 | Not applicable |
| ModbusMaster.Slave2.Main.SearchDevice | Initiates a slave search (0 = No; 1 = Yes) | bool | 7d6e | 32110 | Not applicable |
| ModbusMaster.Slave2.Main.SearchResult | Current search status (as Slave1.Data) | uint8 | 7d70 | 32112 | Not applicable |
| ModbusMaster.Slave2.Main.Timeout | Time in milliseconds the master will wait for a response | float32 | 7b08 | 31496 | 0dp |
| ModbusMaster.Slave2.Main.UnitId | Unit id for a slave device | uint8 | 7b03 | 31491 | Not applicable |
| Mux8.1.Fallback | Fallback Strategy<br>0 = Clip Bad; 1 = Clip Good; 2 = Fallback Good<br>3 = Fallback Good; 4 = Up scale; 5 = Down scale. | uint8 | 2f66 | 12134 | Not applicable |
| Mux8.1.FallbackVal | Fallback Value | float32 | 2f67 | 12135 | 1dp |
| Mux8.1.HighLimit | High Limit | float32 | 2f69 | 12137 | 1dp |
| Mux8.1.In1 | Input 1 | float32 | 2f6b | 12139 | 1dp |
| Mux8.1.In2 | Input 2 | float32 | 2f6c | 12140 | 1dp |
| Mux8.1.In3 | Input 3 | float32 | 2f6d | 12141 | 1dp |
| Mux8.1.In4 | Input 4 | float32 | 2f6e | 12142 | 1dp |
| Mux8.1.In5 | Input 5 | float32 | 2f6f | 12143 | 1dp |
| Mux8.1.In6 | Input 6 | float32 | 2f70 | 12144 | 1dp |
| Mux8.1.In7 | Input 7 | float32 | 2f71 | 12145 | 1dp |
| Mux8.1.In8 | Input 8 | float32 | 2f72 | 12146 | 1dp |
| Mux8.1.LowLimit | Low Limit | float32 | 2f6a | 12138 | 1dp |
| Mux8.1.Out | Output | float32 | 2f73 | 12147 | Set by Mux8.1.Resolution |
| Mux8.1.Resolution | Resolution | uint8 | 2f75 | 12149 | Not applicable |
| Mux8.1.Select | Input Selection Switch<br>1 to 8 = input 1 to 8 (respectively) selected for output | uint8 | 2f68 | 12136 | Not applicable |
| Mux8.1.Status | Status. 0 = Good (OK); 7 = Bad (Error) | bool | 2f74 | 12148 | Not applicable |
| Mux8.2.Fallback | Fallback Strategy (as Mux8.1.Fallback) | uint8 | 2f76 | 12150 | Not applicable |
| Mux8.2.FallbackVal | Fallback Value | float32 | 2f77 | 12151 | 1dp |
| Mux8.2.HighLimit | High Limit | float32 | 2f79 | 12153 | 1dp |
| Mux8.2.In1 | Input 1 | float32 | 2f7b | 12155 | 1dp |
| Mux8.2.In2 | Input 2 | float32 | 2f7c | 12156 | 1dp |
| Mux8.2.In3 | Input 3 | float32 | 2f7d | 12157 | 1dp |
| Mux8.2.In4 | Input 4 | float32 | 2f7e | 12158 | 1dp |
| Mux8.2.In5 | Input 5 | float32 | 2f7f | 12159 | 1dp |
| Mux8.2.In6 | Input 6 | float32 | 2f80 | 12160 | 1dp |
| Mux8.2.In7 | Input 7 | float32 | 2f81 | 12161 | 1dp |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| Mux8.2.In8 | Input 8 | float32 | 2f82 | 12162 | 1dp |
| Mux8.2.LowLimit | Low Limit | float32 | 2f7a | 12154 | 1dp |
| Mux8.2.Out | Output | float32 | 2f83 | 12163 | Set by Mux8.2.Resolution |
| Mux8.2.Resolution | Resolution | uint8 | 2f85 | 12165 | Not applicable |
| Mux8.2.Select | Input Selection (as Mux8.1.Select) | uint8 | 2f78 | 12152 | Not applicable |
| Mux8.2.Status | Status. 0 = Good (OK); 7 = Bad (Error) | bool | 2f84 | 12164 | Not applicable |
| Mux8.3.Fallback | Fallback Strategy (as Mux8.1.Fallback) | uint8 | 2f86 | 12166 | Not applicable |
| Mux8.3.FallbackVal | Fallback Value | float32 | 2f87 | 12167 | 1dp |
| Mux8.3.HighLimit | High Limit | float32 | 2f89 | 12169 | 1dp |
| Mux8.3.In1 | Input 1 | float32 | 2f8b | 12171 | 1dp |
| Mux8.3.In2 | Input 2 | float32 | 2f8c | 12172 | 1dp |
| Mux8.3.In3 | Input 3 | float32 | 2f8d | 12173 | 1dp |
| Mux8.3.In4 | Input 4 | float32 | 2f8e | 12174 | 1dp |
| Mux8.3.In5 | Input 5 | float32 | 2f8f | 12175 | 1dp |
| Mux8.3.In6 | Input 6 | float32 | 2f90 | 12176 | 1dp |
| Mux8.3.In7 | Input 7 | float32 | 2f91 | 12177 | 1dp |
| Mux8.3.In8 | Input 8 | float32 | 2f92 | 12178 | 1dp |
| Mux8.3.LowLimit | Low Limit | float32 | 2f8a | 12170 | 1dp |
| Mux8.3.Out | Output | float32 | 2f93 | 12179 | Set by Mux8.3.Resolution |
| Mux8.3.Resolution | Resolution | uint8 | 2f95 | 12181 | Not applicable |
| Mux8.3.Select | Input Selection (as Mux8.1.Select) | uint8 | 2f88 | 12168 | Not applicable |
| Mux8.3.Status | Status. 0 = Good (OK); 7 = Bad (Error) | bool | 2f94 | 12180 | Not applicable |
| Mux8.4.Fallback | Fallback Strategy (as Mux8.1.Fallback) | uint8 | 2f96 | 12182 | Not applicable |
| Mux8.4.FallbackVal | Fallback Value | float32 | 2f97 | 12183 | 1dp |
| Mux8.4.HighLimit | High Limit | float32 | 2f99 | 12185 | 1dp |
| Mux8.4.In1 | Input 1 | float32 | 2f9b | 12187 | 1dp |
| Mux8.4.In2 | Input 2 | float32 | 2f9c | 12188 | 1dp |
| Mux8.4.In3 | Input 3 | float32 | 2f9d | 12189 | 1dp |
| Mux8.4.In4 | Input 4 | float32 | 2f9e | 12190 | 1dp |
| Mux8.4.In5 | Input 5 | float32 | 2f9f | 12191 | 1dp |
| Mux8.4.In6 | Input 6 | float32 | 2fa0 | 12192 | 1dp |
| Mux8.4.In7 | Input 7 | float32 | 2fa1 | 12193 | 1dp |
| Mux8.4.In8 | Input 8 | float32 | 2fa2 | 12194 | 1dp |
| Mux8.4.LowLimit | Low Limit | float32 | 2f9a | 12186 | 1dp |
| Mux8.4.Out | Output | float32 | 2fa3 | 12195 | Set by Mux8.4.Resolution |
| Mux8.4.Resolution | Resolution | uint8 | 2fa5 | 12197 | Not applicable |
| Mux8.4.Select | Input Selection (as Mux8.1.Select) | uint8 | 2f98 | 12184 | Not applicable |
| Mux8.4.Status | Status. 0 = Good (OK); 7 = Bad (Error) | bool | 2fa4 | 12196 | Not applicable |
| nano_ui.Access | Access level<br>0 = Logged out; 1 = Operator; 2 = Supervisor; 3 = Engineer | uint8 | 2c00 | 11264 | Not applicable |
| nano_ui.Password | Password | string_t | 5400 | 21504 | Not applicable |
| Network.Archive.ArchiveRate | Rate at which to archive history files<br>0 = None 1 = Every minute 2 = Hourly<br>3 = Daily 4 = Weekly 5 = Monthly<br>6 = Automatic | uint8 | 1114 | 4372 | Not applicable |
| Network.Archive.CSVDateFormat | Date/Time format (0 = Text; 1 = spreadsheet numeric) | uint8 | 111d | 4381 | Not applicable |
| Network.Archive.CSVHeaders | Include header details (0 = No; 1 = Yes) | bool | 111b | 4379 | Not applicable |
| Network.Archive.CSVHeadings | Include headings (0 = No; 1 = Yes) | bool | 111c | 4380 | Not applicable |
| Network.Archive.CSVIncludeValues | Include process values (0 = No; 1 = Yes) | bool | 1119 | 4377 | Not applicable |
| Network.Archive.CSVMessages | Include messages (0 = No; 1 = Yes) | bool | 111a | 4378 | Not applicable |
| Network.Archive.CSVTabDelimiter | Use Tab delimeter instead of comma (0 = No; 1 = Yes) | bool | 111e | 4382 | Not applicable |
| Network.Archive.Destination | Archive destination. 0 = USB; 1 = FTP Server | uint8 | 1111 | 4369 | Not applicable |
| Network.Archive.FileFormat | Archive file format (0 = Binary; 1 = CSV; 2 = both) | uint8 | 1115 | 4373 | Not applicable |
| Network.Archive.MediaDuration | Time in days until the USB is full | float32 | 1118 | 4376 | 2dp |
| Network.Interface.Gateway | Default gateway internet protocol address | string_t | 4524 | 17700 | Not applicable |
| Network.Interface.IPaddress | Internet Protocol (IP) address of this instrument | string_t | 4500 | 17664 | Not applicable |
| Network.Interface.IPType | IP Lookup. 0 = DHCP, 1 = Fixed | uint8 | 1102 | 4354 | Not applicable |
| Network.Interface.MAC | Media Access Control (MAC) address of this instrument | string_t | 4548 | 17736 | Not applicable |
| Network.Interface.SubnetMask | Sub network identification mask | string_t | 4512 | 17682 | Not applicable |
| Network.Modbus.Address | Modbus address for this instrument | uint8 | 1140 | 4416 | Not applicable |
| Network.Modbus.InputTimeout | Modbus Input inactivity timeout (in seconds) | int16 | 1141 | 4417 | Not applicable |
| Network.Modbus.PrefMasterIP | Preferred master IP | string_t | 469c | 18076 | Not applicable |
| Network.Modbus.SerialMode | Modbus serial port mode | uint8 | 1143 | 4419 | Not applicable |
| Network.Modbus.TimeFormat | Time parameter comms resolution | uint8 | 1144 | 4420 | Not applicable |
| Network.Modbus.UnitIdEnable | Unit ident enable | uint8 | 1142 | 4418 | Not applicable |
| OR.1.Input1 | OR Block 1, input 1. 0 = off; 1 = on | bool | 2d00 | 11520 | Not applicable |
| OR.1.Input2 | OR Block 1, input 2. 0 = off; 1 = on | bool | 2d01 | 11521 | Not applicable |
| OR.1.Input3 | OR Block 1, input 3. 0 = off; 1 = on | bool | 2d02 | 11522 | Not applicable |
| OR.1.Input4 | OR Block 1, input 4. 0 = off; 1 = on | bool | 2d03 | 11523 | Not applicable |
| OR.1.Input5 | OR Block 1, input 5. 0 = off; 1 = on | bool | 2d04 | 11524 | Not applicable |
| OR.1.Input6 | OR Block 1, input 6. 0 = off; 1 = on | bool | 2d05 | 11525 | Not applicable |
| OR.1.Input7 | OR Block 1, input 7. 0 = off; 1 = on | bool | 2d06 | 11526 | Not applicable |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| OR.1.Input8 | OR Block 1, input 8. 0 = off; 1 = on | bool | 2d07 | 11527 | Not applicable |
| OR.1.Output | OR Block 1, output. 0 = off; 1 = on | bool | 2d08 | 11528 | Not applicable |
| OR.2.Input1 | OR Block 2, input 1. 0 = off; 1 = on | bool | 2d10 | 11536 | Not applicable |
| OR.2.Input2 | OR Block 2, input 2. 0 = off; 1 = on | bool | 2d11 | 11537 | Not applicable |
| OR.2.Input3 | OR Block 2, input 3. 0 = off; 1 = on | bool | 2d12 | 11538 | Not applicable |
| OR.2.Input4 | OR Block 2, input 4. 0 = off; 1 = on | bool | 2d13 | 11539 | Not applicable |
| OR.2.Input5 | OR Block 2, input 5. 0 = off; 1 = on | bool | 2d14 | 11540 | Not applicable |
| OR.2.Input6 | OR Block 2, input 6. 0 = off; 1 = on | bool | 2d15 | 11541 | Not applicable |
| OR.2.Input7 | OR Block 2, input 7. 0 = off; 1 = on | bool | 2d16 | 11542 | Not applicable |
| OR.2.Input8 | OR Block 2, input 8. 0 = off; 1 = on | bool | 2d17 | 11543 | Not applicable |
| OR.2.Output | OR Block 1, output. 0 = off; 1 = on | bool | 2d18 | 11544 | Not applicable |
| OR.3.Input1 | OR Block 3, input 1. 0 = off; 1 = on | bool | 2d20 | 11552 | Not applicable |
| OR.3.Input2 | OR Block 3, input 2. 0 = off; 1 = on | bool | 2d21 | 11553 | Not applicable |
| OR.3.Input3 | OR Block 3, input 3. 0 = off; 1 = on | bool | 2d22 | 11554 | Not applicable |
| OR.3.Input4 | OR Block 3, input 4. 0 = off; 1 = on | bool | 2d23 | 11555 | Not applicable |
| OR.3.Input5 | OR Block 3, input 5. 0 = off; 1 = on | bool | 2d24 | 11556 | Not applicable |
| OR.3.Input6 | OR Block 3, input 6. 0 = off; 1 = on | bool | 2d25 | 11557 | Not applicable |
| OR.3.Input7 | OR Block 3, input 7. 0 = off; 1 = on | bool | 2d26 | 11558 | Not applicable |
| OR.3.Input8 | OR Block 3, input 8. 0 = off; 1 = on | bool | 2d27 | 11559 | Not applicable |
| OR.3.Output | OR Block 3, output. 0 = off; 1 = on | bool | 2d28 | 11560 | Not applicable |
| OR.4.Input1 | OR Block 4, input 1. 0 = off; 1 = on | bool | 2d30 | 11568 | Not applicable |
| OR.4.Input2 | OR Block 4, input 2. 0 = off; 1 = on | bool | 2d31 | 11569 | Not applicable |
| OR.4.Input3 | OR Block 4, input 3. 0 = off; 1 = on | bool | 2d32 | 11570 | Not applicable |
| OR.4.Input4 | OR Block 4, input 4. 0 = off; 1 = on | bool | 2d33 | 11571 | Not applicable |
| OR.4.Input5 | OR Block 4, input 5. 0 = off; 1 = on | bool | 2d34 | 11572 | Not applicable |
| OR.4.Input6 | OR Block 4, input 6. 0 = off; 1 = on | bool | 2d35 | 11573 | Not applicable |
| OR.4.Input7 | OR Block 4, input 7. 0 = off; 1 = on | bool | 2d36 | 11574 | Not applicable |
| OR.4.Input8 | OR Block 4, input 8. 0 = off; 1 = on | bool | 2d37 | 11575 | Not applicable |
| OR.4.Output | OR Block 4, output. 0 = off; 1 = on | bool | 2d38 | 11576 | Not applicable |
| OR.5.Input1 | OR Block 5, input 1. 0 = off; 1 = on | bool | 2d40 | 11584 | Not applicable |
| OR.5.Input2 | OR Block 5, input 2. 0 = off; 1 = on | bool | 2d41 | 11585 | Not applicable |
| OR.5.Input3 | OR Block 5, input 3. 0 = off; 1 = on | bool | 2d42 | 11586 | Not applicable |
| OR.5.Input4 | OR Block 5, input 4. 0 = off; 1 = on | bool | 2d43 | 11587 | Not applicable |
| OR.5.Input5 | OR Block 5, input 5. 0 = off; 1 = on | bool | 2d44 | 11588 | Not applicable |
| OR.5.Input6 | OR Block 5, input 6. 0 = off; 1 = on | bool | 2d45 | 11589 | Not applicable |
| OR.5.Input7 | OR Block 5, input 7. 0 = off; 1 = on | bool | 2d46 | 11590 | Not applicable |
| OR.5.Input8 | OR Block 5, input 8. 0 = off; 1 = on | bool | 2d47 | 11591 | Not applicable |
| OR.5.Output | OR Block 5, output. 0 = off; 1 = on | bool | 2d48 | 11592 | Not applicable |
| OR.6.Input1 | OR Block 6, input 1. 0 = off; 1 = on | bool | 2d50 | 11600 | Not applicable |
| OR.6.Input2 | OR Block 6, input 2. 0 = off; 1 = on | bool | 2d51 | 11601 | Not applicable |
| OR.6.Input3 | OR Block 6, input 3. 0 = off; 1 = on | bool | 2d52 | 11602 | Not applicable |
| OR.6.Input4 | OR Block 6, input 4. 0 = off; 1 = on | bool | 2d53 | 11603 | Not applicable |
| OR.6.Input5 | OR Block 6, input 5. 0 = off; 1 = on | bool | 2d54 | 11604 | Not applicable |
| OR.6.Input6 | OR Block 6, input 6. 0 = off; 1 = on | bool | 2d55 | 11605 | Not applicable |
| OR.6.Input7 | OR Block 6, input 7. 0 = off; 1 = on | bool | 2d56 | 11606 | Not applicable |
| OR.6.Input8 | OR Block 6, input 8. 0 = off; 1 = on | bool | 2d57 | 11607 | Not applicable |
| OR.6.Output | OR Block 6, output. 0 = off; 1 = on | bool | 2d58 | 11608 | Not applicable |
| OR.7.Input1 | OR Block 7, input 1. 0 = off; 1 = on | bool | 2d60 | 11616 | Not applicable |
| OR.7.Input2 | OR Block 7, input 2. 0 = off; 1 = on | bool | 2d61 | 11617 | Not applicable |
| OR.7.Input3 | OR Block 7, input 3. 0 = off; 1 = on | bool | 2d62 | 11618 | Not applicable |
| OR.7.Input4 | OR Block 7, input 4. 0 = off; 1 = on | bool | 2d63 | 11619 | Not applicable |
| OR.7.Input5 | OR Block 7, input 5. 0 = off; 1 = on | bool | 2d64 | 11620 | Not applicable |
| OR.7.Input6 | OR Block 7, input 6. 0 = off; 1 = on | bool | 2d65 | 11621 | Not applicable |
| OR.7.Input7 | OR Block 7, input 7. 0 = off; 1 = on | bool | 2d66 | 11622 | Not applicable |
| OR.7.Input8 | OR Block 7, input 8. 0 = off; 1 = on | bool | 2d67 | 11623 | Not applicable |
| OR.7.Output | OR Block 7, output. 0 = off; 1 = on | bool | 2d68 | 11624 | Not applicable |
| OR.8.Input1 | OR Block 8, input 1. 0 = off; 1 = on | bool | 2d70 | 11632 | Not applicable |
| OR.8.Input2 | OR Block 8, input 2. 0 = off; 1 = on | bool | 2d71 | 11633 | Not applicable |
| OR.8.Input3 | OR Block 8, input 3. 0 = off; 1 = on | bool | 2d72 | 11634 | Not applicable |
| OR.8.Input4 | OR Block 8, input 4. 0 = off; 1 = on | bool | 2d73 | 11635 | Not applicable |
| OR.8.Input5 | OR Block 8, input 5. 0 = off; 1 = on | bool | 2d74 | 11636 | Not applicable |
| OR.8.Input6 | OR Block 8, input 6. 0 = off; 1 = on | bool | 2d75 | 11637 | Not applicable |
| OR.8.Input7 | OR Block 8, input 7. 0 = off; 1 = on | bool | 2d76 | 11638 | Not applicable |
| OR.8.Input8 | OR Block 8, input 8. 0 = off; 1 = on | bool | 2d77 | 11639 | Not applicable |
| OR.8.Output | OR Block 8, output. 0 = off; 1 = on | bool | 2d78 | 11640 | Not applicable |
| OR.9.Input1 | OR Block 9, input 1. 0 = off; 1 = on | bool | 2d80 | 11648 | Not applicable |
| OR.9.Input2 | OR Block 9, input 2. 0 = off; 1 = on | bool | 2d81 | 11649 | Not applicable |
| OR.9.Input3 | OR Block 9, input 3. 0 = off; 1 = on | bool | 2d82 | 11650 | Not applicable |
| OR.9.Input4 | OR Block 9, input 4. 0 = off; 1 = on | bool | 2d83 | 11651 | Not applicable |
| OR.9.Input5 | OR Block 9, input 5. 0 = off; 1 = on | bool | 2d84 | 11652 | Not applicable |
| OR.9.Input6 | OR Block 9, input 6. 0 = off; 1 = on | bool | 2d85 | 11653 | Not applicable |
| OR.9.Input7 | OR Block 9, input 7. 0 = off; 1 = on | bool | 2d86 | 11654 | Not applicable |
| OR.9.Input8 | OR Block 9, input 8. 0 = off; 1 = on | bool | 2d87 | 11655 | Not applicable |
| OR.9.Output | OR Block 9, output. 0 = off; 1 = on | bool | 2d88 | 11656 | Not applicable |
| OR.10.Input1 | OR Block 10, input 1. 0 = off; 1 = on | bool | 2d90 | 11664 | Not applicable |
| OR.10.Input2 | OR Block 10, input 2. 0 = off; 1 = on | bool | 2d91 | 11665 | Not applicable |
| OR.10.Input3 | OR Block 10, input 3. 0 = off; 1 = on | bool | 2d92 | 11666 | Not applicable |
| OR.10.Input4 | OR Block 10, input 4. 0 = off; 1 = on | bool | 2d93 | 11667 | Not applicable |
| OR.10.Input5 | OR Block 10, input 5. 0 = off; 1 = on | bool | 2d94 | 11668 | Not applicable |
| OR.10.Input6 | OR Block 10, input 6. 0 = off; 1 = on | bool | 2d95 | 11669 | Not applicable |
| OR.10.Input7 | OR Block 10, input 7. 0 = off; 1 = on | bool | 2d96 | 11670 | Not applicable |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| OR.10.Input8 | OR Block 10, input 8. 0 = off; 1 = on | bool | 2d97 | 11671 | Not applicable |
| OR.10.Output | OR Block 10, output. 0 = off; 1 = on | bool | 2d98 | 11672 | Not applicable |
| OR.11.Input1 | OR Block 11, input 1. 0 = off; 1 = on | bool | 2da0 | 11680 | Not applicable |
| OR.11.Input2 | OR Block 11, input 2. 0 = off; 1 = on | bool | 2da1 | 11681 | Not applicable |
| OR.11.Input3 | OR Block 11, input 3. 0 = off; 1 = on | bool | 2da2 | 11682 | Not applicable |
| OR.11.Input4 | OR Block 11, input 4. 0 = off; 1 = on | bool | 2da3 | 11683 | Not applicable |
| OR.11.Input5 | OR Block 11, input 5. 0 = off; 1 = on | bool | 2da4 | 11684 | Not applicable |
| OR.11.Input6 | OR Block 11, input 6. 0 = off; 1 = on | bool | 2da5 | 11685 | Not applicable |
| OR.11.Input7 | OR Block 11, input 7. 0 = off; 1 = on | bool | 2da6 | 11686 | Not applicable |
| OR.11.Input8 | OR Block 11, input 8. 0 = off; 1 = on | bool | 2da7 | 11687 | Not applicable |
| OR.11.Output | OR Block 11, output. 0 = off; 1 = on | bool | 2da8 | 11688 | Not applicable |
| OR.12.Input1 | OR Block 12, input 1. 0 = off; 1 = on | bool | 2db0 | 11696 | Not applicable |
| OR.12.Input2 | OR Block 12, input 2. 0 = off; 1 = on | bool | 2db1 | 11697 | Not applicable |
| OR.12.Input3 | OR Block 12, input 3. 0 = off; 1 = on | bool | 2db2 | 11698 | Not applicable |
| OR.12.Input4 | OR Block 12, input 4. 0 = off; 1 = on | bool | 2db3 | 11699 | Not applicable |
| OR.12.Input5 | OR Block 12, input 5. 0 = off; 1 = on | bool | 2db4 | 11700 | Not applicable |
| OR.12.Input6 | OR Block 12, input 6. 0 = off; 1 = on | bool | 2db5 | 11701 | Not applicable |
| OR.12.Input7 | OR Block 12, input 7. 0 = off; 1 = on | bool | 2db6 | 11702 | Not applicable |
| OR.12.Input8 | OR Block 12, input 8. 0 = off; 1 = on | bool | 2db7 | 11703 | Not applicable |
| OR.12.Output | OR Block 12, output. 0 = off; 1 = on | bool | 2db8 | 11704 | Not applicable |
| Program.Ch1Holdback | Channel 1 holdback type<br>0 = Off 1 = Low 2 = High 3 = Band | uint8 | 3aa1 | 15009 | Not applicable |
| Program.Ch1HoldbackVal | Channel 1 holdback value | float32 | 3aa3 | 15011 | Same as Programmer.SetUp.Ch1PVInput |
| Program.Ch1RampUnits | Channel 1 ramp units | uint8 | 3aa6 | 15014 | Not applicable |
| Program.Ch2Holdback | Channel 2 holdback type (as for Program.Ch1, above) | uint8 | 3aa2 | 15010 | Not applicable |
| Program.Ch2HoldbackVal | Channel 2 holdback value | float32 | 3aa4 | 15012 | Same as Programmer.SetUp.Ch2PVInput |
| Program.Ch2RampUnits | Channel 2 ramp units | uint8 | 3aa7 | 15015 | Not applicable |
| Program.HoldbackStyle | Holdback style (0 = per segment; 1 = per program) | uint8 | 3aa0 | 15008 | Not applicable |
| Program.Program | Program | string_t | 6abb | 27323 | Not applicable |
| Program.RampStyle | Ramp style (0 = Time; 1 = Rate) | uint8 | 3aa5 | 15013 | Not applicable |
| Programmer.Features.FTPStore | FTP store feature enable | bool | 3a04 | 14852 | Not applicable |
| Programmer.Features.Holdback | Holdback feature enable | bool | 3a00 | 14848 | Not applicable |
| Programmer.Features.Messages | Messages feature enable | bool | 3a03 | 14851 | Not applicable |
| Programmer.Features.PVEvent | PV Event feature enable | bool | 3a01 | 14849 | Not applicable |
| Programmer.Features.UserValue | User value feature enable | bool | 3a02 | 14850 | Not applicable |
| Programmer.FileList.Filename1 | Filename | string_t | 7900 | 30976 | Not applicable |
| Programmer.FileList.Filename2 | Filename | string_t | 7901 | 30977 | Not applicable |
| Programmer.FileList.Filename3 | Filename | string_t | 7902 | 30978 | Not applicable |
| Programmer.FileList.Filename4 | Filename | string_t | 7903 | 30979 | Not applicable |
| Programmer.FileList.Filename5 | Filename | string_t | 7904 | 30980 | Not applicable |
| Programmer.FileList.Filename6 | Filename | string_t | 7905 | 30981 | Not applicable |
| Programmer.FileList.Filename7 | Filename | string_t | 7906 | 30982 | Not applicable |
| Programmer.FileList.Filename8 | Filename | string_t | 7907 | 30983 | Not applicable |
| Programmer.FileList.Filename9 | Filename | string_t | 7908 | 30984 | Not applicable |
| Programmer.FileList.Filename10 | Filename | string_t | 7909 | 30985 | Not applicable |
| Programmer.FileList.Filename11 | Filename | string_t | 790a | 30986 | Not applicable |
| Programmer.FileList.Filename12 | Filename | string_t | 790b | 30987 | Not applicable |
| Programmer.FileList.Filename13 | Filename | string_t | 790c | 30988 | Not applicable |
| Programmer.FileList.Filename14 | Filename | string_t | 790d | 30989 | Not applicable |
| Programmer.FileList.Filename15 | Filename | string_t | 790e | 30990 | Not applicable |
| Programmer.FileList.Filename16 | Filename | string_t | 790f | 30991 | Not applicable |
| Programmer.FileList.Filename17 | Filename | string_t | 7910 | 30992 | Not applicable |
| Programmer.FileList.Filename18 | Filename | string_t | 7911 | 30993 | Not applicable |
| Programmer.FileList.Filename19 | Filename | string_t | 7912 | 30994 | Not applicable |
| Programmer.FileList.Filename20 | Filename | string_t | 7913 | 30995 | Not applicable |
| Programmer.FileList.Filename21 | Filename | string_t | 7914 | 30996 | Not applicable |
| Programmer.FileList.Filename22 | Filename | string_t | 7915 | 30997 | Not applicable |
| Programmer.FileList.Filename23 | Filename | string_t | 7916 | 30998 | Not applicable |
| Programmer.FileList.Filename24 | Filename | string_t | 7917 | 30999 | Not applicable |
| Programmer.FileList.Filename25 | Filename | string_t | 7918 | 31000 | Not applicable |
| Programmer.FileList.Filename26 | Filename | string_t | 7919 | 31001 | Not applicable |
| Programmer.FileList.Filename27 | Filename | string_t | 791a | 31002 | Not applicable |
| Programmer.FileList.Filename28 | Filename | string_t | 791b | 31003 | Not applicable |
| Programmer.FileList.Filename29 | Filename | string_t | 791c | 31004 | Not applicable |
| Programmer.FileList.Filename30 | Filename | string_t | 791d | 31005 | Not applicable |
| Programmer.FileList.Filename31 | Filename | string_t | 791e | 31006 | Not applicable |
| Programmer.FileList.Filename32 | Filename | string_t | 791f | 31007 | Not applicable |
| Programmer.FileList.Filename33 | Filename | string_t | 7920 | 31008 | Not applicable |
| Programmer.FileList.Filename34 | Filename | string_t | 7921 | 31009 | Not applicable |
| Programmer.FileList.Filename35 | Filename | string_t | 7922 | 31010 | Not applicable |
| Programmer.FileList.Filename36 | Filename | string_t | 7923 | 31011 | Not applicable |
| Programmer.FileList.Filename37 | Filename | string_t | 7924 | 31012 | Not applicable |
| Programmer.FileList.Filename38 | Filename | string_t | 7925 | 31013 | Not applicable |
| Programmer.FileList.Filename39 | Filename | string_t | 7926 | 31014 | Not applicable |
| Programmer.FileList.Filename40 | Filename | string_t | 7927 | 31015 | Not applicable |
| Programmer.FileList.Filename41 | Filename | string_t | 7928 | 31016 | Not applicable |
| Programmer.FileList.Filename42 | Filename | string_t | 7929 | 31017 | Not applicable |
| Programmer.FileList.Filename43 | Filename | string_t | 792a | 31018 | Not applicable |
| Programmer.FileList.Filename44 | Filename | string_t | 792b | 31019 | Not applicable |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| Programmer.FileList.Filename45 | Filename | string_t | 792c | 31020 | Not applicable |
| Programmer.FileList.Filename46 | Filename | string_t | 792d | 31021 | Not applicable |
| Programmer.FileList.Filename47 | Filename | string_t | 792e | 31022 | Not applicable |
| Programmer.FileList.Filename48 | Filename | string_t | 792f | 31023 | Not applicable |
| Programmer.FileList.Filename49 | Filename | string_t | 7930 | 31024 | Not applicable |
| Programmer.FileList.Filename50 | Filename | string_t | 7931 | 31025 | Not applicable |
| Programmer.FileList.Filename51 | Filename | string_t | 7932 | 31026 | Not applicable |
| Programmer.FileList.Filename52 | Filename | string_t | 7933 | 31027 | Not applicable |
| Programmer.FileList.Filename53 | Filename | string_t | 7934 | 31028 | Not applicable |
| Programmer.FileList.Filename54 | Filename | string_t | 7935 | 31029 | Not applicable |
| Programmer.FileList.Filename55 | Filename | string_t | 7936 | 31030 | Not applicable |
| Programmer.FileList.Filename56 | Filename | string_t | 7937 | 31031 | Not applicable |
| Programmer.FileList.Filename57 | Filename | string_t | 7938 | 31032 | Not applicable |
| Programmer.FileList.Filename58 | Filename | string_t | 7939 | 31033 | Not applicable |
| Programmer.FileList.Filename59 | Filename | string_t | 793a | 31034 | Not applicable |
| Programmer.FileList.Filename60 | Filename | string_t | 793b | 31035 | Not applicable |
| Programmer.FileList.Filename61 | Filename | string_t | 793c | 31036 | Not applicable |
| Programmer.FileList.Filename62 | Filename | string_t | 793d | 31037 | Not applicable |
| Programmer.FileList.Filename63 | Filename | string_t | 793e | 31038 | Not applicable |
| Programmer.FileList.Filename64 | Filename | string_t | 793f | 31039 | Not applicable |
| Programmer.FileList.Filename65 | Filename | string_t | 7940 | 31040 | Not applicable |
| Programmer.FileList.Filename66 | Filename | string_t | 7941 | 31041 | Not applicable |
| Programmer.FileList.Filename67 | Filename | string_t | 7942 | 31042 | Not applicable |
| Programmer.FileList.Filename68 | Filename | string_t | 7943 | 31043 | Not applicable |
| Programmer.FileList.Filename69 | Filename | string_t | 7944 | 31044 | Not applicable |
| Programmer.FileList.Filename70 | Filename | string_t | 7945 | 31045 | Not applicable |
| Programmer.FileList.Filename71 | Filename | string_t | 7946 | 31046 | Not applicable |
| Programmer.FileList.Filename72 | Filename | string_t | 7947 | 31047 | Not applicable |
| Programmer.FileList.Filename73 | Filename | string_t | 7948 | 31048 | Not applicable |
| Programmer.FileList.Filename74 | Filename | string_t | 7949 | 31049 | Not applicable |
| Programmer.FileList.Filename75 | Filename | string_t | 794a | 31050 | Not applicable |
| Programmer.FileList.Filename76 | Filename | string_t | 794b | 31051 | Not applicable |
| Programmer.FileList.Filename77 | Filename | string_t | 794c | 31052 | Not applicable |
| Programmer.FileList.Filename78 | Filename | string_t | 794d | 31053 | Not applicable |
| Programmer.FileList.Filename79 | Filename | string_t | 794e | 31054 | Not applicable |
| Programmer.FileList.Filename80 | Filename | string_t | 794f | 31055 | Not applicable |
| Programmer.FileList.Filename81 | Filename | string_t | 7950 | 31056 | Not applicable |
| Programmer.FileList.Filename82 | Filename | string_t | 7951 | 31057 | Not applicable |
| Programmer.FileList.Filename83 | Filename | string_t | 7952 | 31058 | Not applicable |
| Programmer.FileList.Filename84 | Filename | string_t | 7953 | 31059 | Not applicable |
| Programmer.FileList.Filename85 | Filename | string_t | 7954 | 31060 | Not applicable |
| Programmer.FileList.Filename86 | Filename | string_t | 7955 | 31061 | Not applicable |
| Programmer.FileList.Filename87 | Filename | string_t | 7956 | 31062 | Not applicable |
| Programmer.FileList.Filename88 | Filename | string_t | 7957 | 31063 | Not applicable |
| Programmer.FileList.Filename89 | Filename | string_t | 7958 | 31064 | Not applicable |
| Programmer.FileList.Filename90 | Filename | string_t | 7959 | 31065 | Not applicable |
| Programmer.FileList.Filename91 | Filename | string_t | 795a | 31066 | Not applicable |
| Programmer.FileList.Filename92 | Filename | string_t | 795b | 31067 | Not applicable |
| Programmer.FileList.Filename93 | Filename | string_t | 795c | 31068 | Not applicable |
| Programmer.FileList.Filename94 | Filename | string_t | 795d | 31069 | Not applicable |
| Programmer.FileList.Filename95 | Filename | string_t | 795e | 31070 | Not applicable |
| Programmer.FileList.Filename96 | Filename | string_t | 795f | 31071 | Not applicable |
| Programmer.FileList.Filename97 | Filename | string_t | 7960 | 31072 | Not applicable |
| Programmer.FileList.Filename98 | Filename | string_t | 7961 | 31073 | Not applicable |
| Programmer.FileList.Filename99 | Filename | string_t | 7962 | 31074 | Not applicable |
| Programmer.FileList.Filename100 | Filename | string_t | 7963 | 31075 | Not applicable |
| Programmer.FileList.FilenameEntry | Filename of the program to loaded or stored | string_t | 6a91 | 27281 | Not applicable |
| Programmer.FileList.Operation | Operation (0 = Complete; 1 = Get listing; 2 = iTools only) | uint8 | 3a80 | 14976 | Not applicable |
| Programmer.FileList.RefreshList | Refresh list (0 = No; 1 = Yes) | bool | 3a81 | 14977 | Not applicable |
| Programmer.FTP.IPAddress | Internet Protocol address | string_t | 698c | 27020 | Not applicable |
| Programmer.FTP.Password | Password | string_t | 6a2c | 27180 | Not applicable |
| Programmer.FTP.Username | Username | string_t | 6a03 | 27139 | Not applicable |
| Programmer.Run.Ch1PSP | Channel 1 programmer set-point | float32 | 3a53 | 14931 | Same as Programmer.SetUp.Ch1PVInput |
| Programmer.Run.Ch1PVEvent | Channel 1 PV event (0 = Off; 1 = On) | bool | 3a6c | 14956 | Not applicable |
| Programmer.Run.Ch1Rate | Channel 1 rate | float32 | 3a5e | 14942 | Set by Programmer.SetUp.RateResolution |
| Programmer.Run.Ch1Time | Channel 1 time | time_t | 3a5c | 14940 | Set by Network.Modbus.TimeFormat |
| Programmer.Run.Ch1TSP | Channel 1 target set-point | float32 | 3a5a | 14938 | Same as Programmer.SetUp.Ch1PVInput |
| Programmer.Run.Ch1UserVal | Channel 1 user value | float32 | 3a6a | 14954 | 0dp |
| Programmer.Run.Ch2PSP | Channel 2 programmer set-point | float32 | 3a54 | 14932 | Same as Programmer.SetUp.Ch2PVInput |
| Programmer.Run.Ch2PVEvent | Channel 2 PV event (0 = Off; 1 = On) | bool | 3a6d | 14957 | Not applicable |
| Programmer.Run.Ch2Rate | Channel 2 rate | float32 | 3a5f | 14943 | Set by Programmer.SetUp.RateResolution |
| Programmer.Run.Ch2Time | Channel 2 time | time_t | 3a5d | 14941 | Set by Network.Modbus.TimeFormat |
| Programmer.Run.Ch2TSP | Channel 2 target set-point | float32 | 3a5b | 14939 | Same as Programmer.SetUp.Ch2PVInput |
| Programmer.Run.Ch2UserVal | Channel 2 user value | float32 | 3a6b | 14955 | 0dp |
| Programmer.Run.CyclesLeft | Cycles left (-1 = continuous) | int16 | 3a60 | 14944 | Not applicable |
| Programmer.Run.Duration | Duration | time_t | 3a59 | 14937 | Set by Network.Modbus.TimeFormat |
| Programmer.Run.EndOutput | End output (0 = Off; 1 = On) | bool | 3a61 | 14945 | Not applicable |
| Programmer.Run.Event1 | Event 1 (0 = Off; 1 = On) | bool | 3a62 | 14946 | Not applicable |
| Programmer.Run.Event2 | Event 2 (0 = Off; 1 = On) | bool | 3a63 | 14947 | Not applicable |
| Programmer.Run.Event3 | Event 3 (0 = Off; 1 = On) | bool | 3a64 | 14948 | Not applicable |
| Programmer.Run.Event4 | Event 4 (0 = Off; 1 = On) | bool | 3a65 | 14949 | Not applicable |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| Programmer.Run.Event5 | Event 5 (0 = Off; 1 = On) | bool | 3a66 | 14950 | Not applicable |
| Programmer.Run.Event6 | Event 6 (0 = Off; 1 = On) | bool | 3a67 | 14951 | Not applicable |
| Programmer.Run.Event7 | Event 7 (0 = Off; 1 = On) | bool | 3a68 | 14952 | Not applicable |
| Programmer.Run.Event8 | Event 8 (0 = Off; 1 = On) | bool | 3a69 | 14953 | Not applicable |
| Programmer.Run.Intervention | Intervention<br>0 = No Program 1 = None<br>2 = User intervention 4 = PV Event | uint8 | 3a6f | 14959 | Not applicable |
| Programmer.Run.Mode | Mode (1 = Reset; 2 = Run; 4 = Hold) | uint8 | 3a50 | 14928 | Not applicable |
| Programmer.Run.ProgTimeLeft | Program time left | time_t | 3a57 | 14935 | Set by Network.Modbus.TimeFormat |
| Programmer.Run.ProgTimeRunning | Program time running | time_t | 3a70 | 14960 | Set by Network.Modbus.TimeFormat |
| Programmer.Run.ProgTimeSpent | Program time spent | time_t | 3a58 | 14936 | Set by Network.Modbus.TimeFormat |
| Programmer.Run.Segment | Segment | string_t | 6aa6 | 27302 | Not applicable |
| Programmer.Run.SegmentType | Segment type<br>0 = End 1 = Ramp 2 = Dwell<br>3 = Step 4 = Wait 5 = Go back | uint8 | 3a52 | 14930 | Not applicable |
| Programmer.Run.SegTimeLeft | Segment time left | time_t | 3a55 | 14933 | Set by Network.Modbus.TimeFormat |
| Programmer.Run.SegTimeRun | Segment time run | time_t | 3a56 | 14934 | Set by Network.Modbus.TimeFormat |
| Programmer.Run.Status | Status<br>1 = Reset 2 = Running 4 = Holding<br>8 = Holdback 16 = Waiting 32 = Complete | uint8 | 3a51 | 14929 | Not applicable |
| Programmer.SetUp.Advance | Advance (0 = No 1 = Yes) | bool | 3a42 | 14914 | Not applicable |
| Programmer.SetUp.Amended | Amended (0 = No 1 = Yes) | bool | 3a44 | 14916 | Not applicable |
| Programmer.SetUp.Ch1PVInput | Channel 1 PV input | float32 | 3a26 | 14886 | Set by Programmer.SetUp.Ch1Resolution |
| Programmer.SetUp.Ch1Resolution | Channel 1 Resolution | uint8 | 3a46 | 14918 | Not applicable |
| Programmer.SetUp.Ch1ServoTo | Channel 1 servo to (0 = PV; 1 = SP) | uint8 | 3a2a | 14890 | Not applicable |
| Programmer.SetUp.Ch1SPInput | Channel 1 SP input | float32 | 3a28 | 14888 | 0dp |
| Programmer.SetUp.Ch1Units | Channel 1 units | string_t | 6a85 | 27269 | Not applicable |
| Programmer.SetUp.Ch2PVInput | Channel 2 PV input | float32 | 3a27 | 14887 | Set by Programmer.SetUp.Ch2Resolution |
| Programmer.SetUp.Ch2Resolution | Channel 2 Resolution | uint8 | 3a47 | 14919 | Not applicable |
| Programmer.SetUp.Ch2ServoTo | Channel 2 servo to (0 = PV; 1 = SP) | uint8 | 3a2b | 14891 | Not applicable |
| Programmer.SetUp.Ch2SPInput | Channel 2 SP input | float32 | 3a29 | 14889 | 0dp |
| Programmer.SetUp.Ch2Units | Channel 2 units | string_t | 6a8b | 27275 | Not applicable |
| Programmer.SetUp.Channels | Number of channels | uint8 | 3a20 | 14880 | Not applicable |
| Programmer.SetUp.FileErrorStatus | File error status<br>0 = Busy 1 = OK 2 = Load open file<br>3 = Store open file 4 =Delete fail 5 = Copy fail<br>6 = Invalid format 7 = Invalid device 8 = Invalid version<br>9 = Invalid number of channels<br>10 = Parameter write failed<br>11 = Store operation failed to complete<br>12 = Load operation failed to complete<br>13 = Delete operation failed to complete<br>14 = Copy operation failed to complete<br>15 = Invalid filename entered or selected<br>16 = General file operation error<br>17 = Would result in more than the ma.x no. of program files | uint8 | 3a45 | 14917 | Not applicable |
| Programmer.SetUp.Hold | Hold (0 = No 1 = Yes) | bool | 3a39 | 14905 | Not applicable |
| Programmer.SetUp.MaxEvents | Maximum events | uint8 | 3a2d | 14893 | Not applicable |
| Programmer.SetUp.Operation | Operation<br>1 = Select 2 =Load 4 = Store<br>8 = Delete 16 = Delete All 32=Copy<br>64 = Copy All | uint8 | 3a40 | 14912 | Not applicable |
| Programmer.SetUp.PowerFailAction | Power fail action (0 = ramp back; 1 = Reset; 2 = Continue) | uint8 | 3a2c | 14892 | Not applicable |
| Programmer.SetUp.ProgEditAccess | Program edit access level<br>0 = Logged out 1 = Operator<br>2 = Supervisor 3 = Engineer | uint8 | 3a22 | 14882 | Not applicable |
| Programmer.SetUp.ProgModeAccess | Program mode access level (as Program Edit Access, above) | uint8 | 3a21 | 14881 | Not applicable |
| Programmer.SetUp.ProgStoreAccess | Program store access level (as Program Edit Access, above) | uint8 | 3a23 | 14883 | Not applicable |
| Programmer.SetUp.RateResolution | Rate resolution | uint8 | 3a24 | 14884 | Not applicable |
| Programmer.SetUp.Reset | Reset (0 = No 1 = Yes) | bool | 3a3a | 14906 | Not applicable |
| Programmer.SetUp.ResetCh1UserVal | Reset channel 1 user value | float32 | 3a36 | 14902 | 1dp |
| Programmer.SetUp.ResetCh2UserVal | Reset channel 2 user value | float32 | 3a37 | 14903 | 1dp |
| Programmer.SetUp.ResetEvent1 | Reset event 1 (0 = Off, 1 = On) | bool | 3a2e | 14894 | Not applicable |
| Programmer.SetUp.ResetEvent2 | Reset event 2 (0 = Off, 1 = On) | bool | 3a2f | 14895 | Not applicable |
| Programmer.SetUp.ResetEvent3 | Reset event 3 (0 = Off, 1 = On) | bool | 3a30 | 14896 | Not applicable |
| Programmer.SetUp.ResetEvent4 | Reset event 4 (0 = Off, 1 = On) | bool | 3a31 | 14897 | Not applicable |
| Programmer.SetUp.ResetEvent5 | Reset event 5 (0 = Off, 1 = On) | bool | 3a32 | 14898 | Not applicable |
| Programmer.SetUp.ResetEvent6 | Reset event 6 (0 = Off, 1 = On) | bool | 3a33 | 14899 | Not applicable |
| Programmer.SetUp.ResetEvent7 | Reset event 7 (0 = Off, 1 = On) | bool | 3a34 | 14900 | Not applicable |
| Programmer.SetUp.ResetEvent8 | Reset event 8 (0 = Off, 1 = On) | bool | 3a35 | 14901 | Not applicable |
| Programmer.SetUp.Run | Run (0 = No 1 = Yes) | bool | 3a38 | 14904 | Not applicable |
| Programmer.SetUp.RunHold | Run Hold (0 = No 1 = Yes) | bool | 3a3c | 14908 | Not applicable |
| Programmer.SetUp.RunReset | Run Rese (0 = No 1 = Yes)t | bool | 3a3b | 14907 | Not applicable |
| Programmer.SetUp.Status | Status<br>0 = Inactive 1 = Success 2 = Failed<br>3 = Loading 4 = Storing 5 = Deleting<br>6 = Copying | uint8 | 3a41 | 14913 | Not applicable |
| Programmer.SetUp.WaitAnalog1 | Wait analog input 1 | float32 | 3a3e | 14910 | 0dp |
| Programmer.SetUp.WaitAnalog2 | Wait analog input 2 | float32 | 3a3f | 14911 | 0dp |
| Programmer.SetUp.WaitDigital | Wait Digital (0 = Off 1 = On) | bool | 3a3d | 14909 | Not applicable |
| RealTimeEvent.1.Duration | Sets the duration for the event to remain On | time_t | 30e6 | 12518 | Set by Network.Modbus.TimeFormat |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| RealTimeEvent.1.OffDate | Sets the date in the month that the event is to switch off | uint8 | 30e8 | 12520 | Not applicable |
| RealTimeEvent.1.OffDay | Sets the day the the event is to switch Off<br>0 = Sunday 1 = Monday 2 = Tuesday<br>3 = Wednesday 4 = Thursday 5 = Friday<br>6 = Saturday 7 = Monday to Friday<br>8 = Saturday to Sunday<br>9 = Every day | uint8 | 30e9 | 12521 | Not applicable |
| RealTimeEvent.1.OffMonth | The month number when the event is to switch off | uint8 | 30e7 | 12519 | Not applicable |
| RealTimeEvent.1.OffTime | Sets the time that the event is to switch Off | time_t | 30ea | 12522 | Set by Network.Modbus.TimeFormat |
| RealTimeEvent.1.OffType | 0 = Duration; 1 = Time | uint8 | 30e5 | 12517 | Not applicable |
| RealTimeEvent.1.OnDate | Sets the date in the month that the event is to switch on | uint8 | 30e2 | 12514 | Not applicable |
| RealTimeEvent.1.OnDay | Sets the day on which event is to switch on (as 'OffDay', above) | uint8 | 30e3 | 12515 | Not applicable |
| RealTimeEvent.1.OnMonth | The month number when the event is to switch on | uint8 | 30e1 | 12513 | Not applicable |
| RealTimeEvent.1.OnTime | Sets the time that the event is to switch On | time_t | 30e4 | 12516 | Set by Network.Modbus.TimeFormat |
| RealTimeEvent.1.Output | The output from the real time event (0 = Off; 1 = On) | bool | 30eb | 12523 | Not applicable |
| RealTimeEvent.1.Type | Selects the type of Real Time Event<br>0 = Off 1 = Time and Day 2 = Time and Date | uint8 | 30e0 | 12512 | Not applicable |
| RealTimeEvent.2.Duration | Sets the duration for the event to remain On | time_t | 30f6 | 12534 | Set by Network.Modbus.TimeFormat |
| RealTimeEvent.2.OffDate | Sets the date in the month that the event is to switch off | uint8 | 30f8 | 12536 | Not applicable |
| RealTimeEvent.2.OffDay | Sets the day the the event is to switch Off (as for Event 1) | uint8 | 30f9 | 12537 | Not applicable |
| RealTimeEvent.2.OffMonth | Sets the month that the event is to switch off | uint8 | 30f7 | 12535 | Not applicable |
| RealTimeEvent.2.OffTime | Sets the time that the event is to switch Off | time_t | 30fa | 12538 | Set by Network.Modbus.TimeFormat |
| RealTimeEvent.2.OffType | Selects the type that will switch off the event (as for Event 1) | uint8 | 30f5 | 12533 | Not applicable |
| RealTimeEvent.2.OnDate | Sets the date in the month that the event is to switch on | uint8 | 30f2 | 12530 | Not applicable |
| RealTimeEvent.2.OnDay | Sets the day the event is to switch on (as for Event 1) | uint8 | 30f3 | 12531 | Not applicable |
| RealTimeEvent.2.OnMonth | Sets the month that the event is to switch on | uint8 | 30f1 | 12529 | Not applicable |
| RealTimeEvent.2.OnTime | Sets the time that the event is to switch On | time_t | 30f4 | 12532 | Set by Network.Modbus.TimeFormat |
| RealTimeEvent.2.Output | The output from the real time event (0 = Off; 1 = On) | bool | 30fb | 12539 | Not applicable |
| RealTimeEvent.2.Type | Selects the type of Real Time Event<br>0 = Off 1 = Time and Day 2 = Time and Date | uint8 | 30f0 | 12528 | Not applicable |
| Segment.1.Ch1Holdback | Channel 1 holdback type<br>0 = Off 1 = Low 2= High 3 = Band | uint8 | 3ac9 | 15049 | Not applicable |
| Segment.1.Ch1HoldbackVal | Channel 1 holdback value | float32 | 3acb | 15051 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.1.Ch1PVEvent | Channel 1 PV event<br>0 = Off 1 = Absolute High 2 = Absolute Low<br>3 = Deviation High 4 = Deviation Low 5 = Deviation Band | uint8 | 3ad4 | 15060 | Not applicable |
| Segment.1.Ch1PVEventUse | Channel 1 PV event use (0 = Trigger; 1 = Alarm) | bool | 3ae2 | 15074 | Not applicable |
| Segment.1.Ch1PVEventVal | Channel 1 PV event value | float32 | 3ad6 | 15062 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.1.Ch1Rate | Channel 1 rate | float32 | 3ac6 | 15046 | Set by Programmer.SetUp.RateResolution |
| Segment.1.Ch1Time | Channel 1 time | time_t | 3ac4 | 15044 | Set by Network.Modbus.TimeFormat |
| Segment.1.Ch1TSP | Channel 1 target set-point | float32 | 3ac2 | 15042 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.1.Ch1UserVal | Channel 1 user value | float32 | 3ad8 | 15064 | Same as Programmer.SetUp.ResetCh1UserVal |
| Segment.1.Ch1Wait | Channel 1 Wait (Analogue 1 criterion)<br>1= Abs high 2 = Abs low 3 = Dev high 4 = Dev Low | uint8 | 3ace | 15054 | Not applicable |
| Segment.1.Ch1WaitVal | Channel 1 wait value | float32 | 3ad0 | 15056 | Same as Programmer.SetUp.PVWait1 |
| Segment.1.Ch2Holdback | Channel 2 holdback type (as for Ch1Holdback, above) | uint8 | 3aca | 15050 | Not applicable |
| Segment.1.Ch2HoldbackVal | Channel 2 holdback value | float32 | 3acc | 15052 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.1.Ch2PVEvent | Channel 2 PV event (as for Ch1PVEvent, above) | uint8 | 3ad5 | 15061 | Not applicable |
| Segment.1.Ch2PVEventUse | Channel 2 PV event use (as for Ch1PVEventUse, above) | bool | 3ae3 | 15075 | Not applicable |
| Segment.1.Ch2PVEventVal | Channel 2 PV event value | float32 | 3ad7 | 15063 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.1.Ch2Rate | Channel 2 rate | float32 | 3ac7 | 15047 | Set by Programmer.SetUp.RateResolu-tion |
| Segment.1.Ch2Time | Channel 2 time | time_t | 3ac5 | 15045 | Set by Network.Modbus.TimeFormat |
| Segment.1.Ch2TSP | Channel 2 target set-point | float32 | 3ac3 | 15043 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.1.Ch2UserVal | Channel 2 user value | float32 | 3ad9 | 15065 | Same as Program-mer.SetUp.ResetCh2UserVal |
| Segment.1.Ch2Wait | Channel 2 Wait (analogue 2 criterion; as for Ch1Wait, above) | uint8 | 3acf | 15055 | Not applicable |
| Segment.1.Ch2WaitVal | Channel 2 wait value | float32 | 3ad1 | 15057 | Same as Programmer.SetUp.PVWait2 |
| Segment.1.Cycles | Cycles (0 = Continuous) | int16 | 3ad3 | 15059 | Not applicable |
| Segment.1.Duration | Duration | time_t | 3ac1 | 15041 | Set by Network.Modbus.TimeFormat |
| Segment.1.EndType | End type (0 = Dwell; 1 = Reset) | uint8 | 3ac8 | 15048 | Not applicable |
| Segment.1.Event1 | Event 1 (0 = Off; 1 = On) | bool | 3ada | 15066 | Not applicable |
| Segment.1.Event2 | Event 2 (0 = Off; 1 = On) | bool | 3adb | 15067 | Not applicable |
| Segment.1.Event3 | Event 3 (0 = Off; 1 = On) | bool | 3adc | 15068 | Not applicable |
| Segment.1.Event4 | Event 4 (0 = Off; 1 = On) | bool | 3add | 15069 | Not applicable |
| Segment.1.Event5 | Event 5 (0 = Off; 1 = On) | bool | 3ade | 15070 | Not applicable |
| Segment.1.Event6 | Event 6 (0 = Off; 1 = On) | bool | 3adf | 15071 | Not applicable |
| Segment.1.Event7 | Event 7 (0 = Off; 1 = On) | bool | 3ae0 | 15072 | Not applicable |
| Segment.1.Event8 | Event 8 (0 = Off; 1 = On) | bool | 3ae1 | 15073 | Not applicable |
| Segment.1.GoBackTo | Go back to | uint8 | 3ad2 | 15058 | Not applicable |
| Segment.1.SegmentName | Segment name | string_t | 6ad0 | 27344 | Not applicable |
| Segment.1.Type | Type<br>0 = End 1 = Ramp 2 = Dwell<br>3 = Step 4 = Wait 5 = GoBack | uint8 | 3ac0 | 15040 | Not applicable |
| Segment.1.WaitFor | Wait for<br>0 = Digital High 1 = Wait analogue 1<br>2 = Wait analogue 2 3 = Wait analogue 1 and analogue 2 | uint8 | 3acd | 15053 | Not applicable |
| Segment.2.Ch1Holdback | Channel 1 holdback type | uint8 | 3af9 | 15097 | Not applicable |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| Segment.2.Ch1HoldbackVal | Channel 1 holdback value | float32 | 3afb | 15099 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.2.Ch1PVEvent | Channel 1 PV event | uint8 | 3b04 | 15108 | Not applicable |
| Segment.2.Ch1PVEventUse | Channel 1 PV event use | bool | 3b12 | 15122 | Not applicable |
| Segment.2.Ch1PVEventVal | Channel 1 PV event value | float32 | 3b06 | 15110 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.2.Ch1Rate | Channel 1 rate | float32 | 3af6 | 15094 | Set by Programmer.SetUp.RateResolution |
| Segment.2.Ch1Time | Channel 1 time | time_t | 3af4 | 15092 | Set by Network.Modbus.TimeFormat |
| Segment.2.Ch1TSP | Channel 1 target set-point | float32 | 3af2 | 15090 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.2.Ch1UserVal | Channel 1 user value | float32 | 3b08 | 15112 | Same as Programmer.SetUp.ResetCh1UserVal |
| Segment.2.Ch1Wait | Channel 1 Wait | uint8 | 3afe | 15102 | Not applicable |
| Segment.2.Ch1WaitVal | Channel 1 wait value | float32 | 3b00 | 15104 | Same as Programmer.SetUp.PVWait1 |
| Segment.2.Ch2Holdback | Channel 2 holdback type | uint8 | 3afa | 15098 | Not applicable |
| Segment.2.Ch2HoldbackVal | Channel 2 holdback value | float32 | 3afc | 15100 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.2.Ch2PVEvent | Channel 2 PV event | uint8 | 3b05 | 15109 | Not applicable |
| Segment.2.Ch2PVEventUse | Channel 2 PV event use | bool | 3b13 | 15123 | Not applicable |
| Segment.2.Ch2PVEventVal | Channel 2 PV event value | float32 | 3b07 | 15111 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.2.Ch2Rate | Channel 2 rate | float32 | 3af7 | 15095 | Set by Programmer.SetUp.RateResolution |
| Segment.2.Ch2Time | Channel 2 time | time_t | 3af5 | 15093 | Set by Network.Modbus.TimeFormat |
| Segment.2.Ch2TSP | Channel 2 target set-point | float32 | 3af3 | 15091 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.2.Ch2UserVal | Channel 2 user value | float32 | 3b09 | 15113 | Same as Programmer.SetUp.ResetCh2UserVal |
| Segment.2.Ch2Wait | Channel 2 Wait | uint8 | 3aff | 15103 | Not applicable |
| Segment.2.Ch2WaitVal | Channel 2 wait value | float32 | 3b01 | 15105 | Same as Programmer.SetUp.PVWait2 |
| Segment.2.Cycles | Cycles | int16 | 3b03 | 15107 | Not applicable |
| Segment.2.Duration | Duration | time_t | 3af1 | 15089 | Set by Network.Modbus.TimeFormat |
| Segment.2.EndType | End type | uint8 | 3af8 | 15096 | Not applicable |
| Segment.2.Event1 | Event 1 | bool | 3b0a | 15114 | Not applicable |
| Segment.2.Event2 | Event 2 | bool | 3b0b | 15115 | Not applicable |
| Segment.2.Event3 | Event 3 | bool | 3b0c | 15116 | Not applicable |
| Segment.2.Event4 | Event 4 | bool | 3b0d | 15117 | Not applicable |
| Segment.2.Event5 | Event 5 | bool | 3b0e | 15118 | Not applicable |
| Segment.2.Event6 | Event 6 | bool | 3b0f | 15119 | Not applicable |
| Segment.2.Event7 | Event 7 | bool | 3b10 | 15120 | Not applicable |
| Segment.2.Event8 | Event 8 | bool | 3b11 | 15121 | Not applicable |
| Segment.2.GoBackTo | Go back to | uint8 | 3b02 | 15106 | Not applicable |
| Segment.2.SegmentName | Segment name | string_t | 6ae5 | 27365 | Not applicable |
| Segment.2.Type | Type | uint8 | 3af0 | 15088 | Not applicable |
| Segment.2.WaitFor | Wait for | uint8 | 3afd | 15101 | Not applicable |
| Segment.3.Ch1Holdback | Channel 1 holdback type | uint8 | 3b29 | 15145 | Not applicable |
| Segment.3.Ch1HoldbackVal | Channel 1 holdback value | float32 | 3b2b | 15147 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.3.Ch1PVEvent | Channel 1 PV event | uint8 | 3b34 | 15156 | Not applicable |
| Segment.3.Ch1PVEventUse | Channel 1 PV event use | bool | 3b42 | 15170 | Not applicable |
| Segment.3.Ch1PVEventVal | Channel 1 PV event value | float32 | 3b36 | 15158 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.3.Ch1Rate | Channel 1 rate | float32 | 3b26 | 15142 | Set by Programmer.SetUp.RateResolution |
| Segment.3.Ch1Time | Channel 1 time | time_t | 3b24 | 15140 | Set by Network.Modbus.TimeFormat |
| Segment.3.Ch1TSP | Channel 1 target set-point | float32 | 3b22 | 15138 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.3.Ch1UserVal | Channel 1 user value | float32 | 3b38 | 15160 | Same as Programmer.SetUp.ResetCh1UserVal |
| Segment.3.Ch1Wait | Channel 1 Wait | uint8 | 3b2e | 15150 | Not applicable |
| Segment.3.Ch1WaitVal | Channel 1 wait value | float32 | 3b30 | 15152 | Same as Programmer.SetUp.PVWait1 |
| Segment.3.Ch2Holdback | Channel 2 holdback type | uint8 | 3b2a | 15146 | Not applicable |
| Segment.3.Ch2HoldbackVal | Channel 2 holdback value | float32 | 3b2c | 15148 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.3.Ch2PVEvent | Channel 2 PV event | uint8 | 3b35 | 15157 | Not applicable |
| Segment.3.Ch2PVEventUse | Channel 2 PV event use | bool | 3b43 | 15171 | Not applicable |
| Segment.3.Ch2PVEventVal | Channel 2 PV event value | float32 | 3b37 | 15159 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.3.Ch2Rate | Channel 2 rate | float32 | 3b27 | 15143 | Set by Programmer.SetUp.RateResolution |
| Segment.3.Ch2Time | Channel 2 time | time_t | 3b25 | 15141 | Set by Network.Modbus.TimeFormat |
| Segment.3.Ch2TSP | Channel 2 target set-point | float32 | 3b23 | 15139 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.3.Ch2UserVal | Channel 2 user value | float32 | 3b39 | 15161 | Same as Programmer.SetUp.ResetCh2UserVal |
| Segment.3.Ch2Wait | Channel 2 Wait | uint8 | 3b2f | 15151 | Not applicable |
| Segment.3.Ch2WaitVal | Channel 2 wait value | float32 | 3b31 | 15153 | Same as Programmer.SetUp.PVWait2 |
| Segment.3.Cycles | Cycles | int16 | 3b33 | 15155 | Not applicable |
| Segment.3.Duration | Duration | time_t | 3b21 | 15137 | Set by Network.Modbus.TimeFormat |
| Segment.3.EndType | End type | uint8 | 3b28 | 15144 | Not applicable |
| Segment.3.Event1 | Event 1 | bool | 3b3a | 15162 | Not applicable |
| Segment.3.Event2 | Event 2 | bool | 3b3b | 15163 | Not applicable |
| Segment.3.Event3 | Event 3 | bool | 3b3c | 15164 | Not applicable |
| Segment.3.Event4 | Event 4 | bool | 3b3d | 15165 | Not applicable |
| Segment.3.Event5 | Event 5 | bool | 3b3e | 15166 | Not applicable |
| Segment.3.Event6 | Event 6 | bool | 3b3f | 15167 | Not applicable |
| Segment.3.Event7 | Event 7 | bool | 3b40 | 15168 | Not applicable |
| Segment.3.Event8 | Event 8 | bool | 3b41 | 15169 | Not applicable |
| Segment.3.GoBackTo | Go back to | uint8 | 3b32 | 15154 | Not applicable |
| Segment.3.SegmentName | Segment name | string_t | 6afa | 27386 | Not applicable |
| Segment.3.Type | Type | uint8 | 3b20 | 15136 | Not applicable |
| Segment.3.WaitFor | Wait for | uint8 | 3b2d | 15149 | Not applicable |
| Segment.4.Ch1Holdback | Channel 1 holdback type | uint8 | 3b59 | 15193 | Not applicable |
| Segment.4.Ch1HoldbackVal | Channel 1 holdback value | float32 | 3b5b | 15195 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.4.Ch1PVEvent | Channel 1 PV event | uint8 | 3b64 | 15204 | Not applicable |
| Segment.4.Ch1PVEventUse | Channel 1 PV event use | bool | 3b72 | 15218 | Not applicable |
| Segment.4.Ch1PVEventVal | Channel 1 PV event value | float32 | 3b66 | 15206 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.4.Ch1Rate | Channel 1 rate | float32 | 3b56 | 15190 | Set by Programmer.SetUp.RateResolution |

For parameter values and settings (enumerations), see Segment 1

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| Segment.4.Ch1Time | Channel 1 time | time_t | 3b54 | 15188 | Set by Network.Modbus.TimeFormat |
| Segment.4.Ch1TSP | Channel 1 target set-point | float32 | 3b52 | 15186 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.4.Ch1UserVal | Channel 1 user value | float32 | 3b68 | 15208 | Same as Programmer.SetUp.ResetCh1UserVal |
| Segment.4.Ch1Wait | Channel 1 Wait | uint8 | 3b5e | 15198 | Not applicable |
| Segment.4.Ch1WaitVal | Channel 1 wait value | float32 | 3b60 | 15200 | Same as Programmer.SetUp.PVWait1 |
| Segment.4.Ch2Holdback | Channel 2 holdback type | uint8 | 3b5a | 15194 | Not applicable |
| Segment.4.Ch2HoldbackVal | Channel 2 holdback value | float32 | 3b5c | 15196 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.4.Ch2PVEvent | Channel 2 PV event | uint8 | 3b65 | 15205 | Not applicable |
| Segment.4.Ch2PVEventUse | Channel 2 PV event use | bool | 3b73 | 15219 | Not applicable |
| Segment.4.Ch2PVEventVal | Channel 2 PV event value | float32 | 3b67 | 15207 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.4.Ch2Rate | Channel 2 rate | float32 | 3b57 | 15191 | Set by Programmer.SetUp.RateResolution |
| Segment.4.Ch2Time | Channel 2 time | time_t | 3b55 | 15189 | Set by Network.Modbus.TimeFormat |
| Segment.4.Ch2TSP | Channel 2 target set-point | float32 | 3b53 | 15187 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.4.Ch2UserVal | Channel 2 user value | float32 | 3b69 | 15209 | Same as Programmer.SetUp.ResetCh2UserVal |
| Segment.4.Ch2Wait | Channel 2 Wait | uint8 | 3b5f | 15199 | Not applicable |
| Segment.4.Ch2WaitVal | Channel 2 wait value | float32 | 3b61 | 15201 | Same as Programmer.SetUp.PVWait2 |
| Segment.4.Cycles | Cycles | int16 | 3b63 | 15203 | Not applicable |
| Segment.4.Duration | Duration | time_t | 3b51 | 15185 | Set by Network.Modbus.TimeFormat |
| Segment.4.EndType | End type | uint8 | 3b58 | 15192 | Not applicable |
| Segment.4.Event1 | Event 1 | bool | 3b6a | 15210 | Not applicable |
| Segment.4.Event2 | Event 2 | bool | 3b6b | 15211 | Not applicable |
| Segment.4.Event3 | Event 3 | bool | 3b6c | 15212 | Not applicable |
| Segment.4.Event4 | Event 4 | bool | 3b6d | 15213 | Not applicable |
| Segment.4.Event5 | Event 5 | bool | 3b6e | 15214 | Not applicable |
| Segment.4.Event6 | Event 6 | bool | 3b6f | 15215 | Not applicable |
| Segment.4.Event7 | Event 7 | bool | 3b70 | 15216 | Not applicable |
| Segment.4.Event8 | Event 8 | bool | 3b71 | 15217 | Not applicable |
| Segment.4.GoBackTo | Go back to | uint8 | 3b62 | 15202 | Not applicable |
| Segment.4.SegmentName | Segment name | string_t | 6b0f | 27407 | Not applicable |
| Segment.4.Type | Type | uint8 | 3b50 | 15184 | Not applicable |
| Segment.4.WaitFor | Wait for | uint8 | 3b5d | 15197 | Not applicable |
| | | | | | |
| Segment.5.Ch1Holdback | Channel 1 holdback type | uint8 | 3b89 | 15241 | Not applicable |
| Segment.5.Ch1HoldbackVal | Channel 1 holdback value | float32 | 3b8b | 15243 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.5.Ch1PVEvent | Channel 1 PV event | uint8 | 3b94 | 15252 | Not applicable |
| Segment.5.Ch1PVEventUse | Channel 1 PV event use | bool | 3ba2 | 15266 | Not applicable |
| Segment.5.Ch1PVEventVal | Channel 1 PV event value | float32 | 3b96 | 15254 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.5.Ch1Rate | Channel 1 rate | float32 | 3b86 | 15238 | Set by Programmer.SetUp.RateResolution |
| Segment.5.Ch1Time | Channel 1 time | time_t | 3b84 | 15236 | Set by Network.Modbus.TimeFormat |
| Segment.5.Ch1TSP | Channel 1 target set-point | float32 | 3b82 | 15234 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.5.Ch1UserVal | Channel 1 user value | float32 | 3b98 | 15256 | Same as Programmer.SetUp.ResetCh1UserVal |
| Segment.5.Ch1Wait | Channel 1 Wait | uint8 | 3b8e | 15246 | Not applicable |
| Segment.5.Ch1WaitVal | Channel 1 wait value | float32 | 3b90 | 15248 | Same as Programmer.SetUp.PVWait1 |
| Segment.5.Ch2Holdback | Channel 2 holdback type | uint8 | 3b8a | 15242 | Not applicable |
| Segment.5.Ch2HoldbackVal | Channel 2 holdback value | float32 | 3b8c | 15244 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.5.Ch2PVEvent | Channel 2 PV event | uint8 | 3b95 | 15253 | Not applicable |
| Segment.5.Ch2PVEventUse | Channel 2 PV event use | bool | 3ba3 | 15267 | Not applicable |
| Segment.5.Ch2PVEventVal | Channel 2 PV event value | float32 | 3b97 | 15255 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.5.Ch2Rate | Channel 2 rate | float32 | 3b87 | 15239 | Set by Programmer.SetUp.RateResolution |
| Segment.5.Ch2Time | Channel 2 time | time_t | 3b85 | 15237 | Set by Network.Modbus.TimeFormat |
| Segment.5.Ch2TSP | Channel 2 target set-point | float32 | 3b83 | 15235 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.5.Ch2UserVal | Channel 2 user value | float32 | 3b99 | 15257 | Same as Programmer.SetUp.ResetCh2UserVal |
| Segment.5.Ch2Wait | Channel 2 Wait | uint8 | 3b8f | 15247 | Not applicable |
| Segment.5.Ch2WaitVal | Channel 2 wait value | float32 | 3b91 | 15249 | Same as Programmer.SetUp.PVWait2 |
| Segment.5.Cycles | Cycles | int16 | 3b93 | 15251 | Not applicable |
| Segment.5.Duration | Duration | time_t | 3b81 | 15233 | Set by Network.Modbus.TimeFormat |
| Segment.5.EndType | End type | uint8 | 3b88 | 15240 | Not applicable |
| Segment.5.Event1 | Event 1 | bool | 3b9a | 15258 | Not applicable |
| Segment.5.Event2 | Event 2 | bool | 3b9b | 15259 | Not applicable |
| Segment.5.Event3 | Event 3 | bool | 3b9c | 15260 | Not applicable |
| Segment.5.Event4 | Event 4 | bool | 3b9d | 15261 | Not applicable |
| Segment.5.Event5 | Event 5 | bool | 3b9e | 15262 | Not applicable |
| Segment.5.Event6 | Event 6 | bool | 3b9f | 15263 | Not applicable |
| Segment.5.Event7 | Event 7 | bool | 3ba0 | 15264 | Not applicable |
| Segment.5.Event8 | Event 8 | bool | 3ba1 | 15265 | Not applicable |
| Segment.5.GoBackTo | Go back to | uint8 | 3b92 | 15250 | Not applicable |
| Segment.5.SegmentName | Segment name | string_t | 6b24 | 27428 | Not applicable |
| Segment.5.Type | Type | uint8 | 3b80 | 15232 | Not applicable |
| Segment.5.WaitFor | Wait for | uint8 | 3b8d | 15245 | Not applicable |
| | | | | | |
| Segment.6.Ch1Holdback | Channel 1 holdback type | uint8 | 3bb9 | 15289 | Not applicable |
| Segment.6.Ch1HoldbackVal | Channel 1 holdback value | float32 | 3bbb | 15291 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.6.Ch1PVEvent | Channel 1 PV event | uint8 | 3bc4 | 15300 | Not applicable |
| Segment.6.Ch1PVEventUse | Channel 1 PV event use | bool | 3bd2 | 15314 | Not applicable |
| Segment.6.Ch1PVEventVal | Channel 1 PV event value | float32 | 3bc6 | 15302 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.6.Ch1Rate | Channel 1 rate | float32 | 3bb6 | 15286 | Set by Programmer.SetUp.RateResolution |
| Segment.6.Ch1Time | Channel 1 time | time_t | 3bb4 | 15284 | Set by Network.Modbus.TimeFormat |
| Segment.6.Ch1TSP | Channel 1 target set-point | float32 | 3bb2 | 15282 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.6.Ch1UserVal | Channel 1 user value | float32 | 3bc8 | 15304 | Same as Programmer.SetUp.ResetCh1UserVal |
| Segment.6.Ch1Wait | Channel 1 Wait | uint8 | 3bbe | 15294 | Not applicable |
| Segment.6.Ch1WaitVal | Channel 1 wait value | float32 | 3bc0 | 15296 | Same as Programmer.SetUp.PVWait1 |

For parameter values and settings (enumerations), see Segment 1

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| Segment.6.Ch2Holdback | Channel 2 holdback type | uint8 | 3bba | 15290 | Not applicable |
| Segment.6.Ch2HoldbackVal | Channel 2 holdback value | float32 | 3bbc | 15292 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.6.Ch2PVEvent | Channel 2 PV event | uint8 | 3bc5 | 15301 | Not applicable |
| Segment.6.Ch2PVEventUse | Channel 2 PV event use | bool | 3bd3 | 15315 | Not applicable |
| Segment.6.Ch2PVEventVal | Channel 2 PV event value | float32 | 3bc7 | 15303 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.6.Ch2Rate | Channel 2 rate | float32 | 3bb7 | 15287 | Set by Programmer.SetUp.RateResolution |
| Segment.6.Ch2Time | Channel 2 time | time_t | 3bb5 | 15285 | Set by Network.Modbus.TimeFormat |
| Segment.6.Ch2TSP | Channel 2 target set-point | float32 | 3bb3 | 15283 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.6.Ch2UserVal | Channel 2 user value | float32 | 3bc9 | 15305 | Same as Programmer.SetUp.ResetCh2UserVal |
| Segment.6.Ch2Wait | Channel 2 Wait | uint8 | 3bbf | 15295 | Not applicable |
| Segment.6.Ch2WaitVal | Channel 2 wait value | float32 | 3bc1 | 15297 | Same as Programmer.SetUp.PVWait2 |
| Segment.6.Cycles | Cycles | int16 | 3bc3 | 15299 | Not applicable |
| Segment.6.Duration | Duration | time_t | 3bb1 | 15281 | Set by Network.Modbus.TimeFormat |
| Segment.6.EndType | End type | uint8 | 3bb8 | 15288 | Not applicable |
| Segment.6.Event1 | Event 1 | bool | 3bca | 15306 | Not applicable |
| Segment.6.Event2 | Event 2 | bool | 3bcb | 15307 | Not applicable |
| Segment.6.Event3 | Event 3 | bool | 3bcc | 15308 | Not applicable |
| Segment.6.Event4 | Event 4 | bool | 3bcd | 15309 | Not applicable |
| Segment.6.Event5 | Event 5 | bool | 3bce | 15310 | Not applicable |
| Segment.6.Event6 | Event 6 | bool | 3bcf | 15311 | Not applicable |
| Segment.6.Event7 | Event 7 | bool | 3bd0 | 15312 | Not applicable |
| Segment.6.Event8 | Event 8 | bool | 3bd1 | 15313 | Not applicable |
| Segment.6.GoBackTo | Go back to | uint8 | 3bc2 | 15298 | Not applicable |
| Segment.6.SegmentName | Segment name | string_t | 6b39 | 27449 | Not applicable |
| Segment.6.Type | Type | uint8 | 3bb0 | 15280 | Not applicable |
| Segment.6.WaitFor | Wait for | uint8 | 3bbd | 15293 | Not applicable |
| Segment.7.Ch1Holdback | Channel 1 holdback type | uint8 | 3be9 | 15337 | Not applicable |
| Segment.7.Ch1HoldbackVal | Channel 1 holdback value | float32 | 3beb | 15339 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.7.Ch1PVEvent | Channel 1 PV event | uint8 | 3bf4 | 15348 | Not applicable |
| Segment.7.Ch1PVEventUse | Channel 1 PV event use | bool | 3c02 | 15362 | Not applicable |
| Segment.7.Ch1PVEventVal | Channel 1 PV event value | float32 | 3bf6 | 15350 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.7.Ch1Rate | Channel 1 rate | float32 | 3be6 | 15334 | Set by Programmer.SetUp.RateResolution |
| Segment.7.Ch1Time | Channel 1 time | time_t | 3be4 | 15332 | Set by Network.Modbus.TimeFormat |
| Segment.7.Ch1TSP | Channel 1 target set-point | float32 | 3be2 | 15330 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.7.Ch1UserVal | Channel 1 user value | float32 | 3bf8 | 15352 | Same as Programmer.SetUp.ResetCh1UserVal |
| Segment.7.Ch1Wait | Channel 1 Wait | uint8 | 3bee | 15342 | Not applicable |
| Segment.7.Ch1WaitVal | Channel 1 wait value | float32 | 3bf0 | 15344 | Same as Programmer.SetUp.PVWait1 |
| Segment.7.Ch2Holdback | Channel 2 holdback type | uint8 | 3bea | 15338 | Not applicable |
| Segment.7.Ch2HoldbackVal | Channel 2 holdback value | float32 | 3bec | 15340 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.7.Ch2PVEvent | Channel 2 PV event | uint8 | 3bf5 | 15349 | Not applicable |
| Segment.7.Ch2PVEventUse | Channel 2 PV event use | bool | 3c03 | 15363 | Not applicable |
| Segment.7.Ch2PVEventVal | Channel 2 PV event value | float32 | 3bf7 | 15351 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.7.Ch2Rate | Channel 2 rate | float32 | 3be7 | 15335 | Set by Programmer.SetUp.RateResolution |
| Segment.7.Ch2Time | Channel 2 time | time_t | 3be5 | 15333 | Set by Network.Modbus.TimeFormat |
| Segment.7.Ch2TSP | Channel 2 target set-point | float32 | 3be3 | 15331 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.7.Ch2UserVal | Channel 2 user value | float32 | 3bf9 | 15353 | Same as Programmer.SetUp.ResetCh2UserVal |
| Segment.7.Ch2Wait | Channel 2 Wait | uint8 | 3bef | 15343 | Not applicable |
| Segment.7.Ch2WaitVal | Channel 2 wait value | float32 | 3bf1 | 15345 | Same as Programmer.SetUp.PVWait2 |
| Segment.7.Cycles | Cycles | int16 | 3bf3 | 15347 | Not applicable |
| Segment.7.Duration | Duration | time_t | 3be1 | 15329 | Set by Network.Modbus.TimeFormat |
| Segment.7.EndType | End type | uint8 | 3be8 | 15336 | Not applicable |
| Segment.7.Event1 | Event 1 | bool | 3bfa | 15354 | Not applicable |
| Segment.7.Event2 | Event 2 | bool | 3bfb | 15355 | Not applicable |
| Segment.7.Event3 | Event 3 | bool | 3bfc | 15356 | Not applicable |
| Segment.7.Event4 | Event 4 | bool | 3bfd | 15357 | Not applicable |
| Segment.7.Event5 | Event 5 | bool | 3bfe | 15358 | Not applicable |
| Segment.7.Event6 | Event 6 | bool | 3bff | 15359 | Not applicable |
| Segment.7.Event7 | Event 7 | bool | 3c00 | 15360 | Not applicable |
| Segment.7.Event8 | Event 8 | bool | 3c01 | 15361 | Not applicable |
| Segment.7.GoBackTo | Go back to | uint8 | 3bf2 | 15346 | Not applicable |
| Segment.7.SegmentName | Segment name | string_t | 6b4e | 27470 | Not applicable |
| Segment.7.Type | Type | uint8 | 3be0 | 15328 | Not applicable |
| Segment.7.WaitFor | Wait for | uint8 | 3bed | 15341 | Not applicable |
| Segment.8.Ch1Holdback | Channel 1 holdback type | uint8 | 3c19 | 15385 | Not applicable |
| Segment.8.Ch1HoldbackVal | Channel 1 holdback value | float32 | 3c1b | 15387 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.8.Ch1PVEvent | Channel 1 PV event | uint8 | 3c24 | 15396 | Not applicable |
| Segment.8.Ch1PVEventUse | Channel 1 PV event use | bool | 3c32 | 15410 | Not applicable |
| Segment.8.Ch1PVEventVal | Channel 1 PV event value | float32 | 3c26 | 15398 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.8.Ch1Rate | Channel 1 rate | float32 | 3c16 | 15382 | Set by Programmer.SetUp.RateResolution |
| Segment.8.Ch1Time | Channel 1 time | time_t | 3c14 | 15380 | Set by Network.Modbus.TimeFormat |
| Segment.8.Ch1TSP | Channel 1 target set-point | float32 | 3c12 | 15378 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.8.Ch1UserVal | Channel 1 user value | float32 | 3c28 | 15400 | Same as Programmer.SetUp.ResetCh1UserVal |
| Segment.8.Ch1Wait | Channel 1 Wait | uint8 | 3c1e | 15390 | Not applicable |
| Segment.8.Ch1WaitVal | Channel 1 wait value | float32 | 3c20 | 15392 | Same as Programmer.SetUp.PVWait1 |
| Segment.8.Ch2Holdback | Channel 2 holdback type | uint8 | 3c1a | 15386 | Not applicable |
| Segment.8.Ch2HoldbackVal | Channel 2 holdback value | float32 | 3c1c | 15388 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.8.Ch2PVEvent | Channel 2 PV event | uint8 | 3c25 | 15397 | Not applicable |
| Segment.8.Ch2PVEventUse | Channel 2 PV event use | bool | 3c33 | 15411 | Not applicable |
| Segment.8.Ch2PVEventVal | Channel 2 PV event value | float32 | 3c27 | 15399 | Same as Programmer.SetUp.Ch2PVInput |

For parameter values and settings (enumerations), see Segment 1

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| Segment.8.Ch2Rate | Channel 2 rate | float32 | 3c17 | 15383 | Set by Programmer.SetUp.RateResolution |
| Segment.8.Ch2Time | Channel 2 time | time_t | 3c15 | 15381 | Set by Network.Modbus.TimeFormat |
| Segment.8.Ch2TSP | Channel 2 target set-point | float32 | 3c13 | 15379 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.8.Ch2UserVal | Channel 2 user value | float32 | 3c29 | 15401 | Same as Programmer.SetUp.ResetCh2UserVal |
| Segment.8.Ch2Wait | Channel 2 Wait | uint8 | 3c1f | 15391 | Not applicable |
| Segment.8.Ch2WaitVal | Channel 2 wait value | float32 | 3c21 | 15393 | Same as Programmer.SetUp.PVWait2 |
| Segment.8.Cycles | Cycles | int16 | 3c23 | 15395 | Not applicable |
| Segment.8.Duration | Duration | time_t | 3c11 | 15377 | Set by Network.Modbus.TimeFormat |
| Segment.8.EndType | End type | uint8 | 3c18 | 15384 | Not applicable |
| Segment.8.Event1 | Event 1 | bool | 3c2a | 15402 | Not applicable |
| Segment.8.Event2 | Event 2 | bool | 3c2b | 15403 | Not applicable |
| Segment.8.Event3 | Event 3 | bool | 3c2c | 15404 | Not applicable |
| Segment.8.Event4 | Event 4 | bool | 3c2d | 15405 | Not applicable |
| Segment.8.Event5 | Event 5 | bool | 3c2e | 15406 | Not applicable |
| Segment.8.Event6 | Event 6 | bool | 3c2f | 15407 | Not applicable |
| Segment.8.Event7 | Event 7 | bool | 3c30 | 15408 | Not applicable |
| Segment.8.Event8 | Event 8 | bool | 3c31 | 15409 | Not applicable |
| Segment.8.GoBackTo | Go back to | uint8 | 3c22 | 15394 | Not applicable |
| Segment.8.SegmentName | Segment name | string_t | 6b63 | 27491 | Not applicable |
| Segment.8.Type | Type | uint8 | 3c10 | 15376 | Not applicable |
| Segment.8.WaitFor | Wait for | uint8 | 3c1d | 15389 | Not applicable |
| | | | | | |
| Segment.9.Ch1Holdback | Channel 1 holdback type | uint8 | 3c49 | 15433 | Not applicable |
| Segment.9.Ch1HoldbackVal | Channel 1 holdback value | float32 | 3c4b | 15435 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.9.Ch1PVEvent | Channel 1 PV event | uint8 | 3c54 | 15444 | Not applicable |
| Segment.9.Ch1PVEventUse | Channel 1 PV event use | bool | 3c62 | 15458 | Not applicable |
| Segment.9.Ch1PVEventVal | Channel 1 PV event value | float32 | 3c56 | 15446 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.9.Ch1Rate | Channel 1 rate | float32 | 3c46 | 15430 | Set by Programmer.SetUp.RateResolution |
| Segment.9.Ch1Time | Channel 1 time | time_t | 3c44 | 15428 | Set by Network.Modbus.TimeFormat |
| Segment.9.Ch1TSP | Channel 1 target set-point | float32 | 3c42 | 15426 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.9.Ch1UserVal | Channel 1 user value | float32 | 3c58 | 15448 | Same as Programmer.SetUp.ResetCh1UserVal |
| Segment.9.Ch1Wait | Channel 1 Wait | uint8 | 3c4e | 15438 | Not applicable |
| Segment.9.Ch1WaitVal | Channel 1 wait value | float32 | 3c50 | 15440 | Same as Programmer.SetUp.PVWait1 |
| Segment.9.Ch2Holdback | Channel 2 holdback type | uint8 | 3c4a | 15434 | Not applicable |
| Segment.9.Ch2HoldbackVal | Channel 2 holdback value | float32 | 3c4c | 15436 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.9.Ch2PVEvent | Channel 2 PV event | uint8 | 3c55 | 15445 | Not applicable |
| Segment.9.Ch2PVEventUse | Channel 2 PV event use | bool | 3c63 | 15459 | Not applicable |
| Segment.9.Ch2PVEventVal | Channel 2 PV event value | float32 | 3c57 | 15447 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.9.Ch2Rate | Channel 2 rate | float32 | 3c47 | 15431 | Set by Programmer.SetUp.RateResolution |
| Segment.9.Ch2Time | Channel 2 time | time_t | 3c45 | 15429 | Set by Network.Modbus.TimeFormat |
| Segment.9.Ch2TSP | Channel 2 target set-point | float32 | 3c43 | 15427 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.9.Ch2UserVal | Channel 2 user value | float32 | 3c59 | 15449 | Same as Programmer.SetUp.ResetCh2UserVal |
| Segment.9.Ch2Wait | Channel 2 Wait | uint8 | 3c4f | 15439 | Not applicable |
| Segment.9.Ch2WaitVal | Channel 2 wait value | float32 | 3c51 | 15441 | Same as Programmer.SetUp.PVWait2 |
| Segment.9.Cycles | Cycles | int16 | 3c53 | 15443 | Not applicable |
| Segment.9.Duration | Duration | time_t | 3c41 | 15425 | Set by Network.Modbus.TimeFormat |
| Segment.9.EndType | End type | uint8 | 3c48 | 15432 | Not applicable |
| Segment.9.Event1 | Event 1 | bool | 3c5a | 15450 | Not applicable |
| Segment.9.Event2 | Event 2 | bool | 3c5b | 15451 | Not applicable |
| Segment.9.Event3 | Event 3 | bool | 3c5c | 15452 | Not applicable |
| Segment.9.Event4 | Event 4 | bool | 3c5d | 15453 | Not applicable |
| Segment.9.Event5 | Event 5 | bool | 3c5e | 15454 | Not applicable |
| Segment.9.Event6 | Event 6 | bool | 3c5f | 15455 | Not applicable |
| Segment.9.Event7 | Event 7 | bool | 3c60 | 15456 | Not applicable |
| Segment.9.Event8 | Event 8 | bool | 3c61 | 15457 | Not applicable |
| Segment.9.GoBackTo | Go back to | uint8 | 3c52 | 15442 | Not applicable |
| Segment.9.SegmentName | Segment name | string_t | 6b78 | 27512 | Not applicable |
| Segment.9.Type | Type | uint8 | 3c40 | 15424 | Not applicable |
| Segment.9.WaitFor | Wait for | uint8 | 3c4d | 15437 | Not applicable |
| | | | | | |
| Segment.10.Ch1Holdback | Channel 1 holdback type | uint8 | 3c79 | 15481 | Not applicable |
| Segment.10.Ch1HoldbackVal | Channel 1 holdback value | float32 | 3c7b | 15483 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.10.Ch1PVEvent | Channel 1 PV event | uint8 | 3c84 | 15492 | Not applicable |
| Segment.10.Ch1PVEventUse | Channel 1 PV event use | bool | 3c92 | 15506 | Not applicable |
| Segment.10.Ch1PVEventVal | Channel 1 PV event value | float32 | 3c86 | 15494 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.10.Ch1Rate | Channel 1 rate | float32 | 3c76 | 15478 | Set by Programmer.SetUp.RateResolution |
| Segment.10.Ch1Time | Channel 1 time | time_t | 3c74 | 15476 | Set by Network.Modbus.TimeFormat |
| Segment.10.Ch1TSP | Channel 1 target set-point | float32 | 3c72 | 15474 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.10.Ch1UserVal | Channel 1 user value | float32 | 3c88 | 15496 | Same as Programmer.SetUp.ResetCh1UserVal |
| Segment.10.Ch1Wait | Channel 1 Wait | uint8 | 3c7e | 15486 | Not applicable |
| Segment.10.Ch1WaitVal | Channel 1 wait value | float32 | 3c80 | 15488 | Same as Programmer.SetUp.PVWait1 |
| Segment.10.Ch2Holdback | Channel 2 holdback type | uint8 | 3c7a | 15482 | Not applicable |
| Segment.10.Ch2HoldbackVal | Channel 2 holdback value | float32 | 3c7c | 15484 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.10.Ch2PVEvent | Channel 2 PV event | uint8 | 3c85 | 15493 | Not applicable |
| Segment.10.Ch2PVEventUse | Channel 2 PV event use | bool | 3c93 | 15507 | Not applicable |
| Segment.10.Ch2PVEventVal | Channel 2 PV event value | float32 | 3c87 | 15495 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.10.Ch2Rate | Channel 2 rate | float32 | 3c77 | 15479 | Set by Programmer.SetUp.RateResolution |
| Segment.10.Ch2Time | Channel 2 time | time_t | 3c75 | 15477 | Set by Network.Modbus.TimeFormat |
| Segment.10.Ch2TSP | Channel 2 target set-point | float32 | 3c73 | 15475 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.10.Ch2UserVal | Channel 2 user value | float32 | 3c89 | 15497 | Same as Programmer.SetUp.ResetCh2UserVal |
| Segment.10.Ch2Wait | Channel 2 Wait | uint8 | 3c7f | 15487 | Not applicable |

For parameter values and settings (enumerations), see Segment 1

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| Segment.10.Ch2WaitVal | Channel 2 wait value | float32 | 3c81 | 15489 | Same as Programmer.SetUp.PVWait2 |
| Segment.10.Cycles | Cycles | int16 | 3c83 | 15491 | Not applicable |
| Segment.10.Duration | Duration | time_t | 3c71 | 15473 | Set by Network.Modbus.TimeFormat |
| Segment.10.EndType | End type | uint8 | 3c78 | 15480 | Not applicable |
| Segment.10.Event1 | Event 1 | bool | 3c8a | 15498 | Not applicable |
| Segment.10.Event2 | Event 2 | bool | 3c8b | 15499 | Not applicable |
| Segment.10.Event3 | Event 3 | bool | 3c8c | 15500 | Not applicable |
| Segment.10.Event4 | Event 4 | bool | 3c8d | 15501 | Not applicable |
| Segment.10.Event5 | Event 5 | bool | 3c8e | 15502 | Not applicable |
| Segment.10.Event6 | Event 6 | bool | 3c8f | 15503 | Not applicable |
| Segment.10.Event7 | Event 7 | bool | 3c90 | 15504 | Not applicable |
| Segment.10.Event8 | Event 8 | bool | 3c91 | 15505 | Not applicable |
| Segment.10.GoBackTo | Go back to | uint8 | 3c82 | 15490 | Not applicable |
| Segment.10.SegmentName | Segment name | string_t | 6b8d | 27533 | Not applicable |
| Segment.10.Type | Type | uint8 | 3c70 | 15472 | Not applicable |
| Segment.10.WaitFor | Wait for | uint8 | 3c7d | 15485 | Not applicable |
| Segment.11.Ch1Holdback | Channel 1 holdback type | uint8 | 3ca9 | 15529 | Not applicable |
| Segment.11.Ch1HoldbackVal | Channel 1 holdback value | float32 | 3cab | 15531 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.11.Ch1PVEvent | Channel 1 PV event | uint8 | 3cb4 | 15540 | Not applicable |
| Segment.11.Ch1PVEventUse | Channel 1 PV event use | bool | 3cc2 | 15554 | Not applicable |
| Segment.11.Ch1PVEventVal | Channel 1 PV event value | float32 | 3cb6 | 15542 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.11.Ch1Rate | Channel 1 rate | float32 | 3ca6 | 15526 | Set by Programmer.SetUp.RateResolution |
| Segment.11.Ch1Time | Channel 1 time | time_t | 3ca4 | 15524 | Set by Network.Modbus.TimeFormat |
| Segment.11.Ch1TSP | Channel 1 target set-point | float32 | 3ca2 | 15522 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.11.Ch1UserVal | Channel 1 user value | float32 | 3cb8 | 15544 | Same as Programmer.SetUp.ResetCh1UserVal |
| Segment.11.Ch1Wait | Channel 1 Wait | uint8 | 3cae | 15534 | Not applicable |
| Segment.11.Ch1WaitVal | Channel 1 wait value | float32 | 3cb0 | 15536 | Same as Programmer.SetUp.PVWait1 |
| Segment.11.Ch2Holdback | Channel 2 holdback type | uint8 | 3caa | 15530 | Not applicable |
| Segment.11.Ch2HoldbackVal | Channel 2 holdback value | float32 | 3cac | 15532 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.11.Ch2PVEvent | Channel 2 PV event | uint8 | 3cb5 | 15541 | Not applicable |
| Segment.11.Ch2PVEventUse | Channel 2 PV event use | bool | 3cc3 | 15555 | Not applicable |
| Segment.11.Ch2PVEventVal | Channel 2 PV event value | float32 | 3cb7 | 15543 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.11.Ch2Rate | Channel 2 rate | float32 | 3ca7 | 15527 | Set by Programmer.SetUp.RateResolution |
| Segment.11.Ch2Time | Channel 2 time | time_t | 3ca5 | 15525 | Set by Network.Modbus.TimeFormat |
| Segment.11.Ch2TSP | Channel 2 target set-point | float32 | 3ca3 | 15523 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.11.Ch2UserVal | Channel 2 user value | float32 | 3cb9 | 15545 | Same as Programmer.SetUp.ResetCh2UserVal |
| Segment.11.Ch2Wait | Channel 2 Wait | uint8 | 3caf | 15535 | Not applicable |
| Segment.11.Ch2WaitVal | Channel 2 wait value | float32 | 3cb1 | 15537 | Same as Programmer.SetUp.PVWait2 |
| Segment.11.Cycles | Cycles | int16 | 3cb3 | 15539 | Not applicable |
| Segment.11.Duration | Duration | time_t | 3ca1 | 15521 | Set by Network.Modbus.TimeFormat |
| Segment.11.EndType | End type | uint8 | 3ca8 | 15528 | Not applicable |
| Segment.11.Event1 | Event 1 | bool | 3cba | 15546 | Not applicable |
| Segment.11.Event2 | Event 2 | bool | 3cbb | 15547 | Not applicable |
| Segment.11.Event3 | Event 3 | bool | 3cbc | 15548 | Not applicable |
| Segment.11.Event4 | Event 4 | bool | 3cbd | 15549 | Not applicable |
| Segment.11.Event5 | Event 5 | bool | 3cbe | 15550 | Not applicable |
| Segment.11.Event6 | Event 6 | bool | 3cbf | 15551 | Not applicable |
| Segment.11.Event7 | Event 7 | bool | 3cc0 | 15552 | Not applicable |
| Segment.11.Event8 | Event 8 | bool | 3cc1 | 15553 | Not applicable |
| Segment.11.GoBackTo | Go back to | uint8 | 3cb2 | 15538 | Not applicable |
| Segment.11.SegmentName | Segment name | string_t | 6ba2 | 27554 | Not applicable |
| Segment.11.Type | Type | uint8 | 3ca0 | 15520 | Not applicable |
| Segment.11.WaitFor | Wait for | uint8 | 3cad | 15533 | Not applicable |
| Segment.12.Ch1Holdback | Channel 1 holdback type | uint8 | 3cd9 | 15577 | Not applicable |
| Segment.12.Ch1HoldbackVal | Channel 1 holdback value | float32 | 3cdb | 15579 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.12.Ch1PVEvent | Channel 1 PV event | uint8 | 3ce4 | 15588 | Not applicable |
| Segment.12.Ch1PVEventUse | Channel 1 PV event use | bool | 3cf2 | 15602 | Not applicable |
| Segment.12.Ch1PVEventVal | Channel 1 PV event value | float32 | 3ce6 | 15590 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.12.Ch1Rate | Channel 1 rate | float32 | 3cd6 | 15574 | Set by Programmer.SetUp.RateResolution |
| Segment.12.Ch1Time | Channel 1 time | time_t | 3cd4 | 15572 | Set by Network.Modbus.TimeFormat |
| Segment.12.Ch1TSP | Channel 1 target set-point | float32 | 3cd2 | 15570 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.12.Ch1UserVal | Channel 1 user value | float32 | 3ce8 | 15592 | Same as Programmer.SetUp.ResetCh1UserVal |
| Segment.12.Ch1Wait | Channel 1 Wait | uint8 | 3cde | 15582 | Not applicable |
| Segment.12.Ch1WaitVal | Channel 1 wait value | float32 | 3ce0 | 15584 | Same as Programmer.SetUp.PVWait1 |
| Segment.12.Ch2Holdback | Channel 2 holdback type | uint8 | 3cda | 15578 | Not applicable |
| Segment.12.Ch2HoldbackVal | Channel 2 holdback value | float32 | 3cdc | 15580 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.12.Ch2PVEvent | Channel 2 PV event | uint8 | 3ce5 | 15589 | Not applicable |
| Segment.12.Ch2PVEventUse | Channel 2 PV event use | bool | 3cf3 | 15603 | Not applicable |
| Segment.12.Ch2PVEventVal | Channel 2 PV event value | float32 | 3ce7 | 15591 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.12.Ch2Rate | Channel 2 rate | float32 | 3cd7 | 15575 | Set by Programmer.SetUp.RateResolution |
| Segment.12.Ch2Time | Channel 2 time | time_t | 3cd5 | 15573 | Set by Network.Modbus.TimeFormat |
| Segment.12.Ch2TSP | Channel 2 target set-point | float32 | 3cd3 | 15571 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.12.Ch2UserVal | Channel 2 user value | float32 | 3ce9 | 15593 | Same as Programmer.SetUp.ResetCh2UserVal |
| Segment.12.Ch2Wait | Channel 2 Wait | uint8 | 3cdf | 15583 | Not applicable |
| Segment.12.Ch2WaitVal | Channel 2 wait value | float32 | 3ce1 | 15585 | Same as Programmer.SetUp.PVWait2 |
| Segment.12.Cycles | Cycles | int16 | 3ce3 | 15587 | Not applicable |
| Segment.12.Duration | Duration | time_t | 3cd1 | 15569 | Set by Network.Modbus.TimeFormat |
| Segment.12.EndType | End type | uint8 | 3cd8 | 15576 | Not applicable |
| Segment.12.Event1 | Event 1 | bool | 3cea | 15594 | Not applicable |

For parameter values and settings (enumerations), see Segment 1

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| Segment.12.Event2 | Event 2 | bool | 3ceb | 15595 | Not applicable |
| Segment.12.Event3 | Event 3 | bool | 3cec | 15596 | Not applicable |
| Segment.12.Event4 | Event 4 | bool | 3ced | 15597 | Not applicable |
| Segment.12.Event5 | Event 5 | bool | 3cee | 15598 | Not applicable |
| Segment.12.Event6 | Event 6 | bool | 3cef | 15599 | Not applicable |
| Segment.12.Event7 | Event 7 | bool | 3cf0 | 15600 | Not applicable |
| Segment.12.Event8 | Event 8 | bool | 3cf1 | 15601 | Not applicable |
| Segment.12.GoBackTo | Go back to | uint8 | 3ce2 | 15586 | Not applicable |
| Segment.12.SegmentName | Segment name | string_t | 6bb7 | 27575 | Not applicable |
| Segment.12.Type | Type | uint8 | 3cd0 | 15568 | Not applicable |
| Segment.12.WaitFor | Wait for | uint8 | 3cdd | 15581 | Not applicable |
| | | | | | |
| Segment.13.Ch1Holdback | Channel 1 holdback type | uint8 | 3d09 | 15625 | Not applicable |
| Segment.13.Ch1HoldbackVal | Channel 1 holdback value | float32 | 3d0b | 15627 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.13.Ch1PVEvent | Channel 1 PV event | uint8 | 3d14 | 15636 | Not applicable |
| Segment.13.Ch1PVEventUse | Channel 1 PV event use | bool | 3d22 | 15650 | Not applicable |
| Segment.13.Ch1PVEventVal | Channel 1 PV event value | float32 | 3d16 | 15638 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.13.Ch1Rate | Channel 1 rate | float32 | 3d06 | 15622 | Set by Programmer.SetUp.RateResolution |
| Segment.13.Ch1Time | Channel 1 time | time_t | 3d04 | 15620 | Set by Network.Modbus.TimeFormat |
| Segment.13.Ch1TSP | Channel 1 target set-point | float32 | 3d02 | 15618 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.13.Ch1UserVal | Channel 1 user value | float32 | 3d18 | 15640 | Same as Programmer.SetUp.ResetCh1UserVal |
| Segment.13.Ch1Wait | Channel 1 Wait | uint8 | 3d0e | 15630 | Not applicable |
| Segment.13.Ch1WaitVal | Channel 1 wait value | float32 | 3d10 | 15632 | Same as Programmer.SetUp.PVWait1 |
| Segment.13.Ch2Holdback | Channel 2 holdback type | uint8 | 3d0a | 15626 | Not applicable |
| Segment.13.Ch2HoldbackVal | Channel 2 holdback value | float32 | 3d0c | 15628 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.13.Ch2PVEvent | Channel 2 PV event | uint8 | 3d15 | 15637 | Not applicable |
| Segment.13.Ch2PVEventUse | Channel 2 PV event use | bool | 3d23 | 15651 | Not applicable |
| Segment.13.Ch2PVEventVal | Channel 2 PV event value | float32 | 3d17 | 15639 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.13.Ch2Rate | Channel 2 rate | float32 | 3d07 | 15623 | Set by Programmer.SetUp.RateResolution |
| Segment.13.Ch2Time | Channel 2 time | time_t | 3d05 | 15621 | Set by Network.Modbus.TimeFormat |
| Segment.13.Ch2TSP | Channel 2 target set-point | float32 | 3d03 | 15619 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.13.Ch2UserVal | Channel 2 user value | float32 | 3d19 | 15641 | Same as Programmer.SetUp.ResetCh2UserVal |
| Segment.13.Ch2Wait | Channel 2 Wait | uint8 | 3d0f | 15631 | Not applicable |
| Segment.13.Ch2WaitVal | Channel 2 wait value | float32 | 3d11 | 15633 | Same as Programmer.SetUp.PVWait2 |
| Segment.13.Cycles | Cycles | int16 | 3d13 | 15635 | Not applicable |
| Segment.13.Duration | Duration | time_t | 3d01 | 15617 | Set by Network.Modbus.TimeFormat |
| Segment.13.EndType | End type | uint8 | 3d08 | 15624 | Not applicable |
| Segment.13.Event1 | Event 1 | bool | 3d1a | 15642 | Not applicable |
| Segment.13.Event2 | Event 2 | bool | 3d1b | 15643 | Not applicable |
| Segment.13.Event3 | Event 3 | bool | 3d1c | 15644 | Not applicable |
| Segment.13.Event4 | Event 4 | bool | 3d1d | 15645 | Not applicable |
| Segment.13.Event5 | Event 5 | bool | 3d1e | 15646 | Not applicable |
| Segment.13.Event6 | Event 6 | bool | 3d1f | 15647 | Not applicable |
| Segment.13.Event7 | Event 7 | bool | 3d20 | 15648 | Not applicable |
| Segment.13.Event8 | Event 8 | bool | 3d21 | 15649 | Not applicable |
| Segment.13.GoBackTo | Go back to | uint8 | 3d12 | 15634 | Not applicable |
| Segment.13.SegmentName | Segment name | string_t | 6bcc | 27596 | Not applicable |
| Segment.13.Type | Type | uint8 | 3d00 | 15616 | Not applicable |
| Segment.13.WaitFor | Wait for | uint8 | 3d0d | 15629 | Not applicable |
| | | | | | |
| Segment.14.Ch1Holdback | Channel 1 holdback type | uint8 | 3d39 | 15673 | Not applicable |
| Segment.14.Ch1HoldbackVal | Channel 1 holdback value | float32 | 3d3b | 15675 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.14.Ch1PVEvent | Channel 1 PV event | uint8 | 3d44 | 15684 | Not applicable |
| Segment.14.Ch1PVEventUse | Channel 1 PV event use | bool | 3d52 | 15698 | Not applicable |
| Segment.14.Ch1PVEventVal | Channel 1 PV event value | float32 | 3d46 | 15686 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.14.Ch1Rate | Channel 1 rate | float32 | 3d36 | 15670 | Set by Programmer.SetUp.RateResolution |
| Segment.14.Ch1Time | Channel 1 time | time_t | 3d34 | 15668 | Set by Network.Modbus.TimeFormat |
| Segment.14.Ch1TSP | Channel 1 target set-point | float32 | 3d32 | 15666 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.14.Ch1UserVal | Channel 1 user value | float32 | 3d48 | 15688 | Same as Programmer.SetUp.ResetCh1UserVal |
| Segment.14.Ch1Wait | Channel 1 Wait | uint8 | 3d3e | 15678 | Not applicable |
| Segment.14.Ch1WaitVal | Channel 1 wait value | float32 | 3d40 | 15680 | Same as Programmer.SetUp.PVWait1 |
| Segment.14.Ch2Holdback | Channel 2 holdback type | uint8 | 3d3a | 15674 | Not applicable |
| Segment.14.Ch2HoldbackVal | Channel 2 holdback value | float32 | 3d3c | 15676 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.14.Ch2PVEvent | Channel 2 PV event | uint8 | 3d45 | 15685 | Not applicable |
| Segment.14.Ch2PVEventUse | Channel 2 PV event use | bool | 3d53 | 15699 | Not applicable |
| Segment.14.Ch2PVEventVal | Channel 2 PV event value | float32 | 3d47 | 15687 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.14.Ch2Rate | Channel 2 rate | float32 | 3d37 | 15671 | Set by Programmer.SetUp.RateResolution |
| Segment.14.Ch2Time | Channel 2 time | time_t | 3d35 | 15669 | Set by Network.Modbus.TimeFormat |
| Segment.14.Ch2TSP | Channel 2 target set-point | float32 | 3d33 | 15667 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.14.Ch2UserVal | Channel 2 user value | float32 | 3d49 | 15689 | Same as Programmer.SetUp.ResetCh2UserVal |
| Segment.14.Ch2Wait | Channel 2 Wait | uint8 | 3d3f | 15679 | Not applicable |
| Segment.14.Ch2WaitVal | Channel 2 wait value | float32 | 3d41 | 15681 | Same as Programmer.SetUp.PVWait2 |
| Segment.14.Cycles | Cycles | int16 | 3d43 | 15683 | Not applicable |
| Segment.14.Duration | Duration | time_t | 3d31 | 15665 | Set by Network.Modbus.TimeFormat |
| Segment.14.EndType | End type | uint8 | 3d38 | 15672 | Not applicable |
| Segment.14.Event1 | Event 1 | bool | 3d4a | 15690 | Not applicable |
| Segment.14.Event2 | Event 2 | bool | 3d4b | 15691 | Not applicable |
| Segment.14.Event3 | Event 3 | bool | 3d4c | 15692 | Not applicable |
| Segment.14.Event4 | Event 4 | bool | 3d4d | 15693 | Not applicable |
| Segment.14.Event5 | Event 5 | bool | 3d4e | 15694 | Not applicable |
| Segment.14.Event6 | Event 6 | bool | 3d4f | 15695 | Not applicable |

For parameter values and settings (enumerations), see Segment 1

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| Segment.14.Event7 | Event 7 | bool | 3d50 | 15696 | Not applicable |
| Segment.14.Event8 | Event 8 | bool | 3d51 | 15697 | Not applicable |
| Segment.14.GoBackTo | Go back to | uint8 | 3d42 | 15682 | Not applicable |
| Segment.14.SegmentName | Segment name | string_t | 6be1 | 27617 | Not applicable |
| Segment.14.Type | Type | uint8 | 3d30 | 15664 | Not applicable |
| Segment.14.WaitFor | Wait for | uint8 | 3d3d | 15677 | Not applicable |
| Segment.15.Ch1Holdback | Channel 1 holdback type | uint8 | 3d69 | 15721 | Not applicable |
| Segment.15.Ch1HoldbackVal | Channel 1 holdback value | float32 | 3d6b | 15723 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.15.Ch1PVEvent | Channel 1 PV event | uint8 | 3d74 | 15732 | Not applicable |
| Segment.15.Ch1PVEventUse | Channel 1 PV event use | bool | 3d82 | 15746 | Not applicable |
| Segment.15.Ch1PVEventVal | Channel 1 PV event value | float32 | 3d76 | 15734 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.15.Ch1Rate | Channel 1 rate | float32 | 3d66 | 15718 | Set by Programmer.SetUp.RateResolution |
| Segment.15.Ch1Time | Channel 1 time | time_t | 3d64 | 15716 | Set by Network.Modbus.TimeFormat |
| Segment.15.Ch1TSP | Channel 1 target set-point | float32 | 3d62 | 15714 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.15.Ch1UserVal | Channel 1 user value | float32 | 3d78 | 15736 | Same as Programmer.SetUp.ResetCh1UserVal |
| Segment.15.Ch1Wait | Channel 1 Wait | uint8 | 3d6e | 15726 | Not applicable |
| Segment.15.Ch1WaitVal | Channel 1 wait value | float32 | 3d70 | 15728 | Same as Programmer.SetUp.PVWait1 |
| Segment.15.Ch2Holdback | Channel 2 holdback type | uint8 | 3d6a | 15722 | Not applicable |
| Segment.15.Ch2HoldbackVal | Channel 2 holdback value | float32 | 3d6c | 15724 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.15.Ch2PVEvent | Channel 2 PV event | uint8 | 3d75 | 15733 | Not applicable |
| Segment.15.Ch2PVEventUse | Channel 2 PV event use | bool | 3d83 | 15747 | Not applicable |
| Segment.15.Ch2PVEventVal | Channel 2 PV event value | float32 | 3d77 | 15735 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.15.Ch2Rate | Channel 2 rate | float32 | 3d67 | 15719 | Set by Programmer.SetUp.RateResolution |
| Segment.15.Ch2Time | Channel 2 time | time_t | 3d65 | 15717 | Set by Network.Modbus.TimeFormat |
| Segment.15.Ch2TSP | Channel 2 target set-point | float32 | 3d63 | 15715 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.15.Ch2UserVal | Channel 2 user value | float32 | 3d79 | 15737 | Same as Programmer.SetUp.ResetCh2UserVal |
| Segment.15.Ch2Wait | Channel 2 Wait | uint8 | 3d6f | 15727 | Not applicable |
| Segment.15.Ch2WaitVal | Channel 2 wait value | float32 | 3d71 | 15729 | Same as Programmer.SetUp.PVWait2 |
| Segment.15.Cycles | Cycles | int16 | 3d73 | 15731 | Not applicable |
| Segment.15.Duration | Duration | time_t | 3d61 | 15713 | Set by Network.Modbus.TimeFormat |
| Segment.15.EndType | End type | uint8 | 3d68 | 15720 | Not applicable |
| Segment.15.Event1 | Event 1 | bool | 3d7a | 15738 | Not applicable |
| Segment.15.Event2 | Event 2 | bool | 3d7b | 15739 | Not applicable |
| Segment.15.Event3 | Event 3 | bool | 3d7c | 15740 | Not applicable |
| Segment.15.Event4 | Event 4 | bool | 3d7d | 15741 | Not applicable |
| Segment.15.Event5 | Event 5 | bool | 3d7e | 15742 | Not applicable |
| Segment.15.Event6 | Event 6 | bool | 3d7f | 15743 | Not applicable |
| Segment.15.Event7 | Event 7 | bool | 3d80 | 15744 | Not applicable |
| Segment.15.Event8 | Event 8 | bool | 3d81 | 15745 | Not applicable |
| Segment.15.GoBackTo | Go back to | uint8 | 3d72 | 15730 | Not applicable |
| Segment.15.SegmentName | Segment name | string_t | 6bf6 | 27638 | Not applicable |
| Segment.15.Type | Type | uint8 | 3d60 | 15712 | Not applicable |
| Segment.15.WaitFor | Wait for | uint8 | 3d6d | 15725 | Not applicable |
| Segment.16.Ch1Holdback | Channel 1 holdback type | uint8 | 3d99 | 15769 | Not applicable |
| Segment.16.Ch1HoldbackVal | Channel 1 holdback value | float32 | 3d9b | 15771 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.16.Ch1PVEvent | Channel 1 PV event | uint8 | 3da4 | 15780 | Not applicable |
| Segment.16.Ch1PVEventUse | Channel 1 PV event use | bool | 3db2 | 15794 | Not applicable |
| Segment.16.Ch1PVEventVal | Channel 1 PV event value | float32 | 3da6 | 15782 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.16.Ch1Rate | Channel 1 rate | float32 | 3d96 | 15766 | Set by Programmer.SetUp.RateResolution |
| Segment.16.Ch1Time | Channel 1 time | time_t | 3d94 | 15764 | Set by Network.Modbus.TimeFormat |
| Segment.16.Ch1TSP | Channel 1 target set-point | float32 | 3d92 | 15762 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.16.Ch1UserVal | Channel 1 user value | float32 | 3da8 | 15784 | Same as Programmer.SetUp.ResetCh1UserVal |
| Segment.16.Ch1Wait | Channel 1 Wait | uint8 | 3d9e | 15774 | Not applicable |
| Segment.16.Ch1WaitVal | Channel 1 wait value | float32 | 3da0 | 15776 | Same as Programmer.SetUp.PVWait1 |
| Segment.16.Ch2Holdback | Channel 2 holdback type | uint8 | 3d9a | 15770 | Not applicable |
| Segment.16.Ch2HoldbackVal | Channel 2 holdback value | float32 | 3d9c | 15772 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.16.Ch2PVEvent | Channel 2 PV event | uint8 | 3da5 | 15781 | Not applicable |
| Segment.16.Ch2PVEventUse | Channel 2 PV event use | bool | 3db3 | 15795 | Not applicable |
| Segment.16.Ch2PVEventVal | Channel 2 PV event value | float32 | 3da7 | 15783 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.16.Ch2Rate | Channel 2 rate | float32 | 3d97 | 15767 | Set by Programmer.SetUp.RateResolution |
| Segment.16.Ch2Time | Channel 2 time | time_t | 3d95 | 15765 | Set by Network.Modbus.TimeFormat |
| Segment.16.Ch2TSP | Channel 2 target set-point | float32 | 3d93 | 15763 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.16.Ch2UserVal | Channel 2 user value | float32 | 3da9 | 15785 | Same as Programmer.SetUp.ResetCh2UserVal |
| Segment.16.Ch2Wait | Channel 2 Wait | uint8 | 3d9f | 15775 | Not applicable |
| Segment.16.Ch2WaitVal | Channel 2 wait value | float32 | 3da1 | 15777 | Same as Programmer.SetUp.PVWait2 |
| Segment.16.Cycles | Cycles | int16 | 3da3 | 15779 | Not applicable |
| Segment.16.Duration | Duration | time_t | 3d91 | 15761 | Set by Network.Modbus.TimeFormat |
| Segment.16.EndType | End type | uint8 | 3d98 | 15768 | Not applicable |
| Segment.16.Event1 | Event 1 | bool | 3daa | 15786 | Not applicable |
| Segment.16.Event2 | Event 2 | bool | 3dab | 15787 | Not applicable |
| Segment.16.Event3 | Event 3 | bool | 3dac | 15788 | Not applicable |
| Segment.16.Event4 | Event 4 | bool | 3dad | 15789 | Not applicable |
| Segment.16.Event5 | Event 5 | bool | 3dae | 15790 | Not applicable |
| Segment.16.Event6 | Event 6 | bool | 3daf | 15791 | Not applicable |
| Segment.16.Event7 | Event 7 | bool | 3db0 | 15792 | Not applicable |
| Segment.16.Event8 | Event 8 | bool | 3db1 | 15793 | Not applicable |
| Segment.16.GoBackTo | Go back to | uint8 | 3da2 | 15778 | Not applicable |
| Segment.16.SegmentName | Segment name | string_t | 6c0b | 27659 | Not applicable |
| Segment.16.Type | Type | uint8 | 3d90 | 15760 | Not applicable |

For parameter values and settings (enumerations), see Segment 1

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| Segment.16.WaitFor | Wait for | uint8 | 3d9d | 15773 | Not applicable |
| Segment.17.Ch1Holdback | Channel 1 holdback type | uint8 | 3dc9 | 15817 | Not applicable |
| Segment.17.Ch1HoldbackVal | Channel 1 holdback value | float32 | 3dcb | 15819 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.17.Ch1PVEvent | Channel 1 PV event | uint8 | 3dd4 | 15828 | Not applicable |
| Segment.17.Ch1PVEventUse | Channel 1 PV event use | bool | 3de2 | 15842 | Not applicable |
| Segment.17.Ch1PVEventVal | Channel 1 PV event value | float32 | 3dd6 | 15830 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.17.Ch1Rate | Channel 1 rate | float32 | 3dc6 | 15814 | Set by Programmer.SetUp.RateResolution |
| Segment.17.Ch1Time | Channel 1 time | time_t | 3dc4 | 15812 | Set by Network.Modbus.TimeFormat |
| Segment.17.Ch1TSP | Channel 1 target set-point | float32 | 3dc2 | 15810 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.17.Ch1UserVal | Channel 1 user value | float32 | 3dd8 | 15832 | Same as Programmer.SetUp.ResetCh1UserVal |
| Segment.17.Ch1Wait | Channel 1 Wait | uint8 | 3dce | 15822 | Not applicable |
| Segment.17.Ch1WaitVal | Channel 1 wait value | float32 | 3dd0 | 15824 | Same as Programmer.SetUp.PVWait1 |
| Segment.17.Ch2Holdback | Channel 2 holdback type | uint8 | 3dca | 15818 | Not applicable |
| Segment.17.Ch2HoldbackVal | Channel 2 holdback value | float32 | 3dcc | 15820 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.17.Ch2PVEvent | Channel 2 PV event | uint8 | 3dd5 | 15829 | Not applicable |
| Segment.17.Ch2PVEventUse | Channel 2 PV event use | bool | 3de3 | 15843 | Not applicable |
| Segment.17.Ch2PVEventVal | Channel 2 PV event value | float32 | 3dd7 | 15831 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.17.Ch2Rate | Channel 2 rate | float32 | 3dc7 | 15815 | Set by Programmer.SetUp.RateResolution |
| Segment.17.Ch2Time | Channel 2 time | time_t | 3dc5 | 15813 | Set by Network.Modbus.TimeFormat |
| Segment.17.Ch2TSP | Channel 2 target set-point | float32 | 3dc3 | 15811 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.17.Ch2UserVal | Channel 2 user value | float32 | 3dd9 | 15833 | Same as Programmer.SetUp.ResetCh2UserVal |
| Segment.17.Ch2Wait | Channel 2 Wait | uint8 | 3dcf | 15823 | Not applicable |
| Segment.17.Ch2WaitVal | Channel 2 wait value | float32 | 3dd1 | 15825 | Same as Programmer.SetUp.PVWait2 |
| Segment.17.Cycles | Cycles | int16 | 3dd3 | 15827 | Not applicable |
| Segment.17.Duration | Duration | time_t | 3dc1 | 15809 | Set by Network.Modbus.TimeFormat |
| Segment.17.EndType | End type | uint8 | 3dc8 | 15816 | Not applicable |
| Segment.17.Event1 | Event 1 | bool | 3dda | 15834 | Not applicable |
| Segment.17.Event2 | Event 2 | bool | 3ddb | 15835 | Not applicable |
| Segment.17.Event3 | Event 3 | bool | 3ddc | 15836 | Not applicable |
| Segment.17.Event4 | Event 4 | bool | 3ddd | 15837 | Not applicable |
| Segment.17.Event5 | Event 5 | bool | 3dde | 15838 | Not applicable |
| Segment.17.Event6 | Event 6 | bool | 3ddf | 15839 | Not applicable |
| Segment.17.Event7 | Event 7 | bool | 3de0 | 15840 | Not applicable |
| Segment.17.Event8 | Event 8 | bool | 3de1 | 15841 | Not applicable |
| Segment.17.GoBackTo | Go back to | uint8 | 3dd2 | 15826 | Not applicable |
| Segment.17.SegmentName | Segment name | string_t | 6c20 | 27680 | Not applicable |
| Segment.17.Type | Type | uint8 | 3dc0 | 15808 | Not applicable |
| Segment.17.WaitFor | Wait for | uint8 | 3dcd | 15821 | Not applicable |
| Segment.18.Ch1Holdback | Channel 1 holdback type | uint8 | 3df9 | 15865 | Not applicable |
| Segment.18.Ch1HoldbackVal | Channel 1 holdback value | float32 | 3dfb | 15867 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.18.Ch1PVEvent | Channel 1 PV event | uint8 | 3e04 | 15876 | Not applicable |
| Segment.18.Ch1PVEventUse | Channel 1 PV event use | bool | 3e12 | 15890 | Not applicable |
| Segment.18.Ch1PVEventVal | Channel 1 PV event value | float32 | 3e06 | 15878 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.18.Ch1Rate | Channel 1 rate | float32 | 3df6 | 15862 | Set by Programmer.SetUp.RateResolution |
| Segment.18.Ch1Time | Channel 1 time | time_t | 3df4 | 15860 | Set by Network.Modbus.TimeFormat |
| Segment.18.Ch1TSP | Channel 1 target set-point | float32 | 3df2 | 15858 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.18.Ch1UserVal | Channel 1 user value | float32 | 3e08 | 15880 | Same as Programmer.SetUp.ResetCh1UserVal |
| Segment.18.Ch1Wait | Channel 1 Wait | uint8 | 3dfe | 15870 | Not applicable |
| Segment.18.Ch1WaitVal | Channel 1 wait value | float32 | 3e00 | 15872 | Same as Programmer.SetUp.PVWait1 |
| Segment.18.Ch2Holdback | Channel 2 holdback type | uint8 | 3dfa | 15866 | Not applicable |
| Segment.18.Ch2HoldbackVal | Channel 2 holdback value | float32 | 3dfc | 15868 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.18.Ch2PVEvent | Channel 2 PV event | uint8 | 3e05 | 15877 | Not applicable |
| Segment.18.Ch2PVEventUse | Channel 2 PV event use | bool | 3e13 | 15891 | Not applicable |
| Segment.18.Ch2PVEventVal | Channel 2 PV event value | float32 | 3e07 | 15879 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.18.Ch2Rate | Channel 2 rate | float32 | 3df7 | 15863 | Set by Programmer.SetUp.RateResolution |
| Segment.18.Ch2Time | Channel 2 time | time_t | 3df5 | 15861 | Set by Network.Modbus.TimeFormat |
| Segment.18.Ch2TSP | Channel 2 target set-point | float32 | 3df3 | 15859 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.18.Ch2UserVal | Channel 2 user value | float32 | 3e09 | 15881 | Same as Programmer.SetUp.ResetCh2UserVal |
| Segment.18.Ch2Wait | Channel 2 Wait | uint8 | 3dff | 15871 | Not applicable |
| Segment.18.Ch2WaitVal | Channel 2 wait value | float32 | 3e01 | 15873 | Same as Programmer.SetUp.PVWait2 |
| Segment.18.Cycles | Cycles | int16 | 3e03 | 15875 | Not applicable |
| Segment.18.Duration | Duration | time_t | 3df1 | 15857 | Set by Network.Modbus.TimeFormat |
| Segment.18.EndType | End type | uint8 | 3df8 | 15864 | Not applicable |
| Segment.18.Event1 | Event 1 | bool | 3e0a | 15882 | Not applicable |
| Segment.18.Event2 | Event 2 | bool | 3e0b | 15883 | Not applicable |
| Segment.18.Event3 | Event 3 | bool | 3e0c | 15884 | Not applicable |
| Segment.18.Event4 | Event 4 | bool | 3e0d | 15885 | Not applicable |
| Segment.18.Event5 | Event 5 | bool | 3e0e | 15886 | Not applicable |
| Segment.18.Event6 | Event 6 | bool | 3e0f | 15887 | Not applicable |
| Segment.18.Event7 | Event 7 | bool | 3e10 | 15888 | Not applicable |
| Segment.18.Event8 | Event 8 | bool | 3e11 | 15889 | Not applicable |
| Segment.18.GoBackTo | Go back to | uint8 | 3e02 | 15874 | Not applicable |
| Segment.18.SegmentName | Segment name | string_t | 6c35 | 27701 | Not applicable |
| Segment.18.Type | Type | uint8 | 3df0 | 15856 | Not applicable |
| Segment.18.WaitFor | Wait for | uint8 | 3dfd | 15869 | Not applicable |
| Segment.19.Ch1Holdback | Channel 1 holdback type | uint8 | 3e29 | 15913 | Not applicable |
| Segment.19.Ch1HoldbackVal | Channel 1 holdback value | float32 | 3e2b | 15915 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.19.Ch1PVEvent | Channel 1 PV event | uint8 | 3e34 | 15924 | Not applicable |

For parameter values and settings (enumerations), see Segment 1

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| Segment.19.Ch1PVEventUse | Channel 1 PV event use | bool | 3e42 | 15938 | Not applicable |
| Segment.19.Ch1PVEventVal | Channel 1 PV event value | float32 | 3e36 | 15926 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.19.Ch1Rate | Channel 1 rate | float32 | 3e26 | 15910 | Set by Programmer.SetUp.RateResolution |
| Segment.19.Ch1Time | Channel 1 time | time_t | 3e24 | 15908 | Set by Network.Modbus.TimeFormat |
| Segment.19.Ch1TSP | Channel 1 target set-point | float32 | 3e22 | 15906 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.19.Ch1UserVal | Channel 1 user value | float32 | 3e38 | 15928 | Same as Programmer.SetUp.ResetCh1UserVal |
| Segment.19.Ch1Wait | Channel 1 Wait | uint8 | 3e2e | 15918 | Not applicable |
| Segment.19.Ch1WaitVal | Channel 1 wait value | float32 | 3e30 | 15920 | Same as Programmer.SetUp.PVWait1 |
| Segment.19.Ch2Holdback | Channel 2 holdback type | uint8 | 3e2a | 15914 | Not applicable |
| Segment.19.Ch2HoldbackVal | Channel 2 holdback value | float32 | 3e2c | 15916 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.19.Ch2PVEvent | Channel 2 PV event | uint8 | 3e35 | 15925 | Not applicable |
| Segment.19.Ch2PVEventUse | Channel 2 PV event use | bool | 3e43 | 15939 | Not applicable |
| Segment.19.Ch2PVEventVal | Channel 2 PV event value | float32 | 3e37 | 15927 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.19.Ch2Rate | Channel 2 rate | float32 | 3e27 | 15911 | Set by Programmer.SetUp.RateResolution |
| Segment.19.Ch2Time | Channel 2 time | time_t | 3e25 | 15909 | Set by Network.Modbus.TimeFormat |
| Segment.19.Ch2TSP | Channel 2 target set-point | float32 | 3e23 | 15907 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.19.Ch2UserVal | Channel 2 user value | float32 | 3e39 | 15929 | Same as Programmer.SetUp.ResetCh2UserVal |
| Segment.19.Ch2Wait | Channel 2 Wait | uint8 | 3e2f | 15919 | Not applicable |
| Segment.19.Ch2WaitVal | Channel 2 wait value | float32 | 3e31 | 15921 | Same as Programmer.SetUp.PVWait2 |
| Segment.19.Cycles | Cycles | int16 | 3e33 | 15923 | Not applicable |
| Segment.19.Duration | Duration | time_t | 3e21 | 15905 | Set by Network.Modbus.TimeFormat |
| Segment.19.EndType | End type | uint8 | 3e28 | 15912 | Not applicable |
| Segment.19.Event1 | Event 1 | bool | 3e3a | 15930 | Not applicable |
| Segment.19.Event2 | Event 2 | bool | 3e3b | 15931 | Not applicable |
| Segment.19.Event3 | Event 3 | bool | 3e3c | 15932 | Not applicable |
| Segment.19.Event4 | Event 4 | bool | 3e3d | 15933 | Not applicable |
| Segment.19.Event5 | Event 5 | bool | 3e3e | 15934 | Not applicable |
| Segment.19.Event6 | Event 6 | bool | 3e3f | 15935 | Not applicable |
| Segment.19.Event7 | Event 7 | bool | 3e40 | 15936 | Not applicable |
| Segment.19.Event8 | Event 8 | bool | 3e41 | 15937 | Not applicable |
| Segment.19.GoBackTo | Go back to | uint8 | 3e32 | 15922 | Not applicable |
| Segment.19.SegmentName | Segment name | string_t | 6c4a | 27722 | Not applicable |
| Segment.19.Type | Type | uint8 | 3e20 | 15904 | Not applicable |
| Segment.19.WaitFor | Wait for | uint8 | 3e2d | 15917 | Not applicable |
| Segment.20.Ch1Holdback | Channel 1 holdback type | uint8 | 3e59 | 15961 | Not applicable |
| Segment.20.Ch1HoldbackVal | Channel 1 holdback value | float32 | 3e5b | 15963 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.20.Ch1PVEvent | Channel 1 PV event | uint8 | 3e64 | 15972 | Not applicable |
| Segment.20.Ch1PVEventUse | Channel 1 PV event use | bool | 3e72 | 15986 | Not applicable |
| Segment.20.Ch1PVEventVal | Channel 1 PV event value | float32 | 3e66 | 15974 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.20.Ch1Rate | Channel 1 rate | float32 | 3e56 | 15958 | Set by Programmer.SetUp.RateResolution |
| Segment.20.Ch1Time | Channel 1 time | time_t | 3e54 | 15956 | Set by Network.Modbus.TimeFormat |
| Segment.20.Ch1TSP | Channel 1 target set-point | float32 | 3e52 | 15954 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.20.Ch1UserVal | Channel 1 user value | float32 | 3e68 | 15976 | Same as Programmer.SetUp.ResetCh1UserVal |
| Segment.20.Ch1Wait | Channel 1 Wait | uint8 | 3e5e | 15966 | Not applicable |
| Segment.20.Ch1WaitVal | Channel 1 wait value | float32 | 3e60 | 15968 | Same as Programmer.SetUp.PVWait1 |
| Segment.20.Ch2Holdback | Channel 2 holdback type | uint8 | 3e5a | 15962 | Not applicable |
| Segment.20.Ch2HoldbackVal | Channel 2 holdback value | float32 | 3e5c | 15964 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.20.Ch2PVEvent | Channel 2 PV event | uint8 | 3e65 | 15973 | Not applicable |
| Segment.20.Ch2PVEventUse | Channel 2 PV event use | bool | 3e73 | 15987 | Not applicable |
| Segment.20.Ch2PVEventVal | Channel 2 PV event value | float32 | 3e67 | 15975 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.20.Ch2Rate | Channel 2 rate | float32 | 3e57 | 15959 | Set by Programmer.SetUp.RateResolution |
| Segment.20.Ch2Time | Channel 2 time | time_t | 3e55 | 15957 | Set by Network.Modbus.TimeFormat |
| Segment.20.Ch2TSP | Channel 2 target set-point | float32 | 3e53 | 15955 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.20.Ch2UserVal | Channel 2 user value | float32 | 3e69 | 15977 | Same as Programmer.SetUp.ResetCh2UserVal |
| Segment.20.Ch2Wait | Channel 2 Wait | uint8 | 3e5f | 15967 | Not applicable |
| Segment.20.Ch2WaitVal | Channel 2 wait value | float32 | 3e61 | 15969 | Same as Programmer.SetUp.PVWait2 |
| Segment.20.Cycles | Cycles | int16 | 3e63 | 15971 | Not applicable |
| Segment.20.Duration | Duration | time_t | 3e51 | 15953 | Set by Network.Modbus.TimeFormat |
| Segment.20.EndType | End type | uint8 | 3e58 | 15960 | Not applicable |
| Segment.20.Event1 | Event 1 | bool | 3e6a | 15978 | Not applicable |
| Segment.20.Event2 | Event 2 | bool | 3e6b | 15979 | Not applicable |
| Segment.20.Event3 | Event 3 | bool | 3e6c | 15980 | Not applicable |
| Segment.20.Event4 | Event 4 | bool | 3e6d | 15981 | Not applicable |
| Segment.20.Event5 | Event 5 | bool | 3e6e | 15982 | Not applicable |
| Segment.20.Event6 | Event 6 | bool | 3e6f | 15983 | Not applicable |
| Segment.20.Event7 | Event 7 | bool | 3e70 | 15984 | Not applicable |
| Segment.20.Event8 | Event 8 | bool | 3e71 | 15985 | Not applicable |
| Segment.20.GoBackTo | Go back to | uint8 | 3e62 | 15970 | Not applicable |
| Segment.20.SegmentName | Segment name | string_t | 6c5f | 27743 | Not applicable |
| Segment.20.Type | Type | uint8 | 3e50 | 15952 | Not applicable |
| Segment.20.WaitFor | Wait for | uint8 | 3e5d | 15965 | Not applicable |
| Segment.21.Ch1Holdback | Channel 1 holdback type | uint8 | 3e89 | 16009 | Not applicable |
| Segment.21.Ch1HoldbackVal | Channel 1 holdback value | float32 | 3e8b | 16011 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.21.Ch1PVEvent | Channel 1 PV event | uint8 | 3e94 | 16020 | Not applicable |
| Segment.21.Ch1PVEventUse | Channel 1 PV event use | bool | 3ea2 | 16034 | Not applicable |
| Segment.21.Ch1PVEventVal | Channel 1 PV event value | float32 | 3e96 | 16022 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.21.Ch1Rate | Channel 1 rate | float32 | 3e86 | 16006 | Set by Programmer.SetUp.RateResolution |
| Segment.21.Ch1Time | Channel 1 time | time_t | 3e84 | 16004 | Set by Network.Modbus.TimeFormat |
| Segment.21.Ch1TSP | Channel 1 target set-point | float32 | 3e82 | 16002 | Same as Programmer.SetUp.Ch1PVInput |

For parameter values and settings (enumerations), see Segment 1

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| Segment.21.Ch1UserVal | Channel 1 user value | float32 | 3e98 | 16024 | Same as Programmer.SetUp.ResetCh1UserVal |
| Segment.21.Ch1Wait | Channel 1 Wait | uint8 | 3e8e | 16014 | Not applicable |
| Segment.21.Ch1WaitVal | Channel 1 wait value | float32 | 3e90 | 16016 | Same as Programmer.SetUp.PVWait1 |
| Segment.21.Ch2Holdback | Channel 2 holdback type | uint8 | 3e8a | 16010 | Not applicable |
| Segment.21.Ch2HoldbackVal | Channel 2 holdback value | float32 | 3e8c | 16012 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.21.Ch2PVEvent | Channel 2 PV event | uint8 | 3e95 | 16021 | Not applicable |
| Segment.21.Ch2PVEventUse | Channel 2 PV event use | bool | 3ea3 | 16035 | Not applicable |
| Segment.21.Ch2PVEventVal | Channel 2 PV event value | float32 | 3e97 | 16023 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.21.Ch2Rate | Channel 2 rate | float32 | 3e87 | 16007 | Set by Programmer.SetUp.RateResolution |
| Segment.21.Ch2Time | Channel 2 time | time_t | 3e85 | 16005 | Set by Network.Modbus.TimeFormat |
| Segment.21.Ch2TSP | Channel 2 target set-point | float32 | 3e83 | 16003 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.21.Ch2UserVal | Channel 2 user value | float32 | 3e99 | 16025 | Same as Programmer.SetUp.ResetCh2UserVal |
| Segment.21.Ch2Wait | Channel 2 Wait | uint8 | 3e8f | 16015 | Not applicable |
| Segment.21.Ch2WaitVal | Channel 2 wait value | float32 | 3e91 | 16017 | Same as Programmer.SetUp.PVWait2 |
| Segment.21.Cycles | Cycles | int16 | 3e93 | 16019 | Not applicable |
| Segment.21.Duration | Duration | time_t | 3e81 | 16001 | Set by Network.Modbus.TimeFormat |
| Segment.21.EndType | End type | uint8 | 3e88 | 16008 | Not applicable |
| Segment.21.Event1 | Event 1 | bool | 3e9a | 16026 | Not applicable |
| Segment.21.Event2 | Event 2 | bool | 3e9b | 16027 | Not applicable |
| Segment.21.Event3 | Event 3 | bool | 3e9c | 16028 | Not applicable |
| Segment.21.Event4 | Event 4 | bool | 3e9d | 16029 | Not applicable |
| Segment.21.Event5 | Event 5 | bool | 3e9e | 16030 | Not applicable |
| Segment.21.Event6 | Event 6 | bool | 3e9f | 16031 | Not applicable |
| Segment.21.Event7 | Event 7 | bool | 3ea0 | 16032 | Not applicable |
| Segment.21.Event8 | Event 8 | bool | 3ea1 | 16033 | Not applicable |
| Segment.21.GoBackTo | Go back to | uint8 | 3e92 | 16018 | Not applicable |
| Segment.21.SegmentName | Segment name | string_t | 6c74 | 27764 | Not applicable |
| Segment.21.Type | Type | uint8 | 3e80 | 16000 | Not applicable |
| Segment.21.WaitFor | Wait for | uint8 | 3e8d | 16013 | Not applicable |
| Segment.22.Ch1Holdback | Channel 1 holdback type | uint8 | 3eb9 | 16057 | Not applicable |
| Segment.22.Ch1HoldbackVal | Channel 1 holdback value | float32 | 3ebb | 16059 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.22.Ch1PVEvent | Channel 1 PV event | uint8 | 3ec4 | 16068 | Not applicable |
| Segment.22.Ch1PVEventUse | Channel 1 PV event use | bool | 3ed2 | 16082 | Not applicable |
| Segment.22.Ch1PVEventVal | Channel 1 PV event value | float32 | 3ec6 | 16070 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.22.Ch1Rate | Channel 1 rate | float32 | 3eb6 | 16054 | Set by Programmer.SetUp.RateResolution |
| Segment.22.Ch1Time | Channel 1 time | time_t | 3eb4 | 16052 | Set by Network.Modbus.TimeFormat |
| Segment.22.Ch1TSP | Channel 1 target set-point | float32 | 3eb2 | 16050 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.22.Ch1UserVal | Channel 1 user value | float32 | 3ec8 | 16072 | Same as Programmer.SetUp.ResetCh1UserVal |
| Segment.22.Ch1Wait | Channel 1 Wait | uint8 | 3ebe | 16062 | Not applicable |
| Segment.22.Ch1WaitVal | Channel 1 wait value | float32 | 3ec0 | 16064 | Same as Programmer.SetUp.PVWait1 |
| Segment.22.Ch2Holdback | Channel 2 holdback type | uint8 | 3eba | 16058 | Not applicable |
| Segment.22.Ch2HoldbackVal | Channel 2 holdback value | float32 | 3ebc | 16060 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.22.Ch2PVEvent | Channel 2 PV event | uint8 | 3ec5 | 16069 | Not applicable |
| Segment.22.Ch2PVEventUse | Channel 2 PV event use | bool | 3ed3 | 16083 | Not applicable |
| Segment.22.Ch2PVEventVal | Channel 2 PV event value | float32 | 3ec7 | 16071 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.22.Ch2Rate | Channel 2 rate | float32 | 3eb7 | 16055 | Set by Programmer.SetUp.RateResolution |
| Segment.22.Ch2Time | Channel 2 time | time_t | 3eb5 | 16053 | Set by Network.Modbus.TimeFormat |
| Segment.22.Ch2TSP | Channel 2 target set-point | float32 | 3eb3 | 16051 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.22.Ch2UserVal | Channel 2 user value | float32 | 3ec9 | 16073 | Same as Programmer.SetUp.ResetCh2UserVal |
| Segment.22.Ch2Wait | Channel 2 Wait | uint8 | 3ebf | 16063 | Not applicable |
| Segment.22.Ch2WaitVal | Channel 2 wait value | float32 | 3ec1 | 16065 | Same as Programmer.SetUp.PVWait2 |
| Segment.22.Cycles | Cycles | int16 | 3ec3 | 16067 | Not applicable |
| Segment.22.Duration | Duration | time_t | 3eb1 | 16049 | Set by Network.Modbus.TimeFormat |
| Segment.22.EndType | End type | uint8 | 3eb8 | 16056 | Not applicable |
| Segment.22.Event1 | Event 1 | bool | 3eca | 16074 | Not applicable |
| Segment.22.Event2 | Event 2 | bool | 3ecb | 16075 | Not applicable |
| Segment.22.Event3 | Event 3 | bool | 3ecc | 16076 | Not applicable |
| Segment.22.Event4 | Event 4 | bool | 3ecd | 16077 | Not applicable |
| Segment.22.Event5 | Event 5 | bool | 3ece | 16078 | Not applicable |
| Segment.22.Event6 | Event 6 | bool | 3ecf | 16079 | Not applicable |
| Segment.22.Event7 | Event 7 | bool | 3ed0 | 16080 | Not applicable |
| Segment.22.Event8 | Event 8 | bool | 3ed1 | 16081 | Not applicable |
| Segment.22.GoBackTo | Go back to | uint8 | 3ec2 | 16066 | Not applicable |
| Segment.22.SegmentName | Segment name | string_t | 6c89 | 27785 | Not applicable |
| Segment.22.Type | Type | uint8 | 3eb0 | 16048 | Not applicable |
| Segment.22.WaitFor | Wait for | uint8 | 3ebd | 16061 | Not applicable |
| Segment.23.Ch1Holdback | Channel 1 holdback type | uint8 | 3ee9 | 16105 | Not applicable |
| Segment.23.Ch1HoldbackVal | Channel 1 holdback value | float32 | 3eeb | 16107 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.23.Ch1PVEvent | Channel 1 PV event | uint8 | 3ef4 | 16116 | Not applicable |
| Segment.23.Ch1PVEventUse | Channel 1 PV event use | bool | 3f02 | 16130 | Not applicable |
| Segment.23.Ch1PVEventVal | Channel 1 PV event value | float32 | 3ef6 | 16118 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.23.Ch1Rate | Channel 1 rate | float32 | 3ee6 | 16102 | Set by Programmer.SetUp.RateResolution |
| Segment.23.Ch1Time | Channel 1 time | time_t | 3ee4 | 16100 | Set by Network.Modbus.TimeFormat |
| Segment.23.Ch1TSP | Channel 1 target set-point | float32 | 3ee2 | 16098 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.23.Ch1UserVal | Channel 1 user value | float32 | 3ef8 | 16120 | Same as Programmer.SetUp.ResetCh1UserVal |
| Segment.23.Ch1Wait | Channel 1 Wait | uint8 | 3eee | 16110 | Not applicable |
| Segment.23.Ch1WaitVal | Channel 1 wait value | float32 | 3ef0 | 16112 | Same as Programmer.SetUp.PVWait1 |
| Segment.23.Ch2Holdback | Channel 2 holdback type | uint8 | 3eea | 16106 | Not applicable |
| Segment.23.Ch2HoldbackVal | Channel 2 holdback value | float32 | 3eec | 16108 | Same as Programmer.SetUp.Ch2PVInput |

For parameter values and settings (enumerations), see Segment 1

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| Segment.23.Ch2PVEvent | Channel 2 PV event | uint8 | 3ef5 | 16117 | Not applicable |
| Segment.23.Ch2PVEventUse | Channel 2 PV event use | bool | 3f03 | 16131 | Not applicable |
| Segment.23.Ch2PVEventVal | Channel 2 PV event value | float32 | 3ef7 | 16119 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.23.Ch2Rate | Channel 2 rate | float32 | 3ee7 | 16103 | Set by Programmer.SetUp.RateResolution |
| Segment.23.Ch2Time | Channel 2 time | time_t | 3ee5 | 16101 | Set by Network.Modbus.TimeFormat |
| Segment.23.Ch2TSP | Channel 2 target set-point | float32 | 3ee3 | 16099 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.23.Ch2UserVal | Channel 2 user value | float32 | 3ef9 | 16121 | Same as Programmer.SetUp.ResetCh2UserVal |
| Segment.23.Ch2Wait | Channel 2 Wait | uint8 | 3eef | 16111 | Not applicable |
| Segment.23.Ch2WaitVal | Channel 2 wait value | float32 | 3ef1 | 16113 | Same as Programmer.SetUp.PVWait2 |
| Segment.23.Cycles | Cycles | int16 | 3ef3 | 16115 | Not applicable |
| Segment.23.Duration | Duration | time_t | 3ee1 | 16097 | Set by Network.Modbus.TimeFormat |
| Segment.23.EndType | End type | uint8 | 3ee8 | 16104 | Not applicable |
| Segment.23.Event1 | Event 1 | bool | 3efa | 16122 | Not applicable |
| Segment.23.Event2 | Event 2 | bool | 3efb | 16123 | Not applicable |
| Segment.23.Event3 | Event 3 | bool | 3efc | 16124 | Not applicable |
| Segment.23.Event4 | Event 4 | bool | 3efd | 16125 | Not applicable |
| Segment.23.Event5 | Event 5 | bool | 3efe | 16126 | Not applicable |
| Segment.23.Event6 | Event 6 | bool | 3eff | 16127 | Not applicable |
| Segment.23.Event7 | Event 7 | bool | 3f00 | 16128 | Not applicable |
| Segment.23.Event8 | Event 8 | bool | 3f01 | 16129 | Not applicable |
| Segment.23.GoBackTo | Go back to | uint8 | 3ef2 | 16114 | Not applicable |
| Segment.23.SegmentName | Segment name | string_t | 6c9e | 27806 | Not applicable |
| Segment.23.Type | Type | uint8 | 3ee0 | 16096 | Not applicable |
| Segment.23.WaitFor | Wait for | uint8 | 3eed | 16109 | Not applicable |
| | | | | | |
| Segment.24.Ch1Holdback | Channel 1 holdback type | uint8 | 3f19 | 16153 | Not applicable |
| Segment.24.Ch1HoldbackVal | Channel 1 holdback value | float32 | 3f1b | 16155 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.24.Ch1PVEvent | Channel 1 PV event | uint8 | 3f24 | 16164 | Not applicable |
| Segment.24.Ch1PVEventUse | Channel 1 PV event use | bool | 3f32 | 16178 | Not applicable |
| Segment.24.Ch1PVEventVal | Channel 1 PV event value | float32 | 3f26 | 16166 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.24.Ch1Rate | Channel 1 rate | float32 | 3f16 | 16150 | Set by Programmer.SetUp.RateResolution |
| Segment.24.Ch1Time | Channel 1 time | time_t | 3f14 | 16148 | Set by Network.Modbus.TimeFormat |
| Segment.24.Ch1TSP | Channel 1 target set-point | float32 | 3f12 | 16146 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.24.Ch1UserVal | Channel 1 user value | float32 | 3f28 | 16168 | Same as Programmer.SetUp.ResetCh1UserVal |
| Segment.24.Ch1Wait | Channel 1 Wait | uint8 | 3f1e | 16158 | Not applicable |
| Segment.24.Ch1WaitVal | Channel 1 wait value | float32 | 3f20 | 16160 | Same as Programmer.SetUp.PVWait1 |
| Segment.24.Ch2Holdback | Channel 2 holdback type | uint8 | 3f1a | 16154 | Not applicable |
| Segment.24.Ch2HoldbackVal | Channel 2 holdback value | float32 | 3f1c | 16156 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.24.Ch2PVEvent | Channel 2 PV event | uint8 | 3f25 | 16165 | Not applicable |
| Segment.24.Ch2PVEventUse | Channel 2 PV event use | bool | 3f33 | 16179 | Not applicable |
| Segment.24.Ch2PVEventVal | Channel 2 PV event value | float32 | 3f27 | 16167 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.24.Ch2Rate | Channel 2 rate | float32 | 3f17 | 16151 | Set by Programmer.SetUp.RateResolution |
| Segment.24.Ch2Time | Channel 2 time | time_t | 3f15 | 16149 | Set by Network.Modbus.TimeFormat |
| Segment.24.Ch2TSP | Channel 2 target set-point | float32 | 3f13 | 16147 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.24.Ch2UserVal | Channel 2 user value | float32 | 3f29 | 16169 | Same as Programmer.SetUp.ResetCh2UserVal |
| Segment.24.Ch2Wait | Channel 2 Wait | uint8 | 3f1f | 16159 | Not applicable |
| Segment.24.Ch2WaitVal | Channel 2 wait value | float32 | 3f21 | 16161 | Same as Programmer.SetUp.PVWait2 |
| Segment.24.Cycles | Cycles | int16 | 3f23 | 16163 | Not applicable |
| Segment.24.Duration | Duration | time_t | 3f11 | 16145 | Set by Network.Modbus.TimeFormat |
| Segment.24.EndType | End type | uint8 | 3f18 | 16152 | Not applicable |
| Segment.24.Event1 | Event 1 | bool | 3f2a | 16170 | Not applicable |
| Segment.24.Event2 | Event 2 | bool | 3f2b | 16171 | Not applicable |
| Segment.24.Event3 | Event 3 | bool | 3f2c | 16172 | Not applicable |
| Segment.24.Event4 | Event 4 | bool | 3f2d | 16173 | Not applicable |
| Segment.24.Event5 | Event 5 | bool | 3f2e | 16174 | Not applicable |
| Segment.24.Event6 | Event 6 | bool | 3f2f | 16175 | Not applicable |
| Segment.24.Event7 | Event 7 | bool | 3f30 | 16176 | Not applicable |
| Segment.24.Event8 | Event 8 | bool | 3f31 | 16177 | Not applicable |
| Segment.24.GoBackTo | Go back to | uint8 | 3f22 | 16162 | Not applicable |
| Segment.24.SegmentName | Segment name | string_t | 6cb3 | 27827 | Not applicable |
| Segment.24.Type | Type | uint8 | 3f10 | 16144 | Not applicable |
| Segment.24.WaitFor | Wait for | uint8 | 3f1d | 16157 | Not applicable |
| | | | | | |
| Segment.25.Ch1Holdback | Channel 1 holdback type | uint8 | 3f49 | 16201 | Not applicable |
| Segment.25.Ch1HoldbackVal | Channel 1 holdback value | float32 | 3f4b | 16203 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.25.Ch1PVEvent | Channel 1 PV event | uint8 | 3f54 | 16212 | Not applicable |
| Segment.25.Ch1PVEventUse | Channel 1 PV event use | bool | 3f62 | 16226 | Not applicable |
| Segment.25.Ch1PVEventVal | Channel 1 PV event value | float32 | 3f56 | 16214 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.25.Ch1Rate | Channel 1 rate | float32 | 3f46 | 16198 | Set by Programmer.SetUp.RateResolution |
| Segment.25.Ch1Time | Channel 1 time | time_t | 3f44 | 16196 | Set by Network.Modbus.TimeFormat |
| Segment.25.Ch1TSP | Channel 1 target set-point | float32 | 3f42 | 16194 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.25.Ch1UserVal | Channel 1 user value | float32 | 3f58 | 16216 | Same as Programmer.SetUp.ResetCh1UserVal |
| Segment.25.Ch1Wait | Channel 1 Wait | uint8 | 3f4e | 16206 | Not applicable |
| Segment.25.Ch1WaitVal | Channel 1 wait value | float32 | 3f50 | 16208 | Same as Programmer.SetUp.PVWait1 |
| Segment.25.Ch2Holdback | Channel 2 holdback type | uint8 | 3f4a | 16202 | Not applicable |
| Segment.25.Ch2HoldbackVal | Channel 2 holdback value | float32 | 3f4c | 16204 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.25.Ch2PVEvent | Channel 2 PV event | uint8 | 3f55 | 16213 | Not applicable |
| Segment.25.Ch2PVEventUse | Channel 2 PV event use | bool | 3f63 | 16227 | Not applicable |
| Segment.25.Ch2PVEventVal | Channel 2 PV event value | float32 | 3f57 | 16215 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.25.Ch2Rate | Channel 2 rate | float32 | 3f47 | 16199 | Set by Programmer.SetUp.RateResolution |
| Segment.25.Ch2Time | Channel 2 time | time_t | 3f45 | 16197 | Set by Network.Modbus.TimeFormat |

For parameter values and settings (enumerations), see Segment 1

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| Segment.25.Ch2TSP | Channel 2 target set-point | float32 | 3f43 | 16195 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.25.Ch2UserVal | Channel 2 user value | float32 | 3f59 | 16217 | Same as Programmer.SetUp.ResetCh2UserVal |
| Segment.25.Ch2Wait | Channel 2 Wait | uint8 | 3f4f | 16207 | Not applicable |
| Segment.25.Ch2WaitVal | Channel 2 wait value | float32 | 3f51 | 16209 | Same as Programmer.SetUp.PVWait2 |
| Segment.25.Cycles | Cycles | int16 | 3f53 | 16211 | Not applicable |
| Segment.25.Duration | Duration | time_t | 3f41 | 16193 | Set by Network.Modbus.TimeFormat |
| Segment.25.EndType | End type | uint8 | 3f48 | 16200 | Not applicable |
| Segment.25.Event1 | Event 1 | bool | 3f5a | 16218 | Not applicable |
| Segment.25.Event2 | Event 2 | bool | 3f5b | 16219 | Not applicable |
| Segment.25.Event3 | Event 3 | bool | 3f5c | 16220 | Not applicable |
| Segment.25.Event4 | Event 4 | bool | 3f5d | 16221 | Not applicable |
| Segment.25.Event5 | Event 5 | bool | 3f5e | 16222 | Not applicable |
| Segment.25.Event6 | Event 6 | bool | 3f5f | 16223 | Not applicable |
| Segment.25.Event7 | Event 7 | bool | 3f60 | 16224 | Not applicable |
| Segment.25.Event8 | Event 8 | bool | 3f61 | 16225 | Not applicable |
| Segment.25.GoBackTo | Go back to | uint8 | 3f52 | 16210 | Not applicable |
| Segment.25.SegmentName | Segment name | string_t | 6cc8 | 27848 | Not applicable |
| Segment.25.Type | Type | uint8 | 3f40 | 16192 | Not applicable |
| Segment.25.WaitFor | Wait for | uint8 | 3f4d | 16205 | Not applicable |
| Segment.26.Ch1Holdback | Channel 1 holdback type | uint8 | 3f79 | 16249 | Not applicable |
| Segment.26.Ch1HoldbackVal | Channel 1 holdback value | float32 | 3f7b | 16251 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.26.Ch1PVEvent | Channel 1 PV event | uint8 | 3f84 | 16260 | Not applicable |
| Segment.26.Ch1PVEventUse | Channel 1 PV event use | bool | 3f92 | 16274 | Not applicable |
| Segment.26.Ch1PVEventVal | Channel 1 PV event value | float32 | 3f86 | 16262 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.26.Ch1Rate | Channel 1 rate | float32 | 3f76 | 16246 | Set by Programmer.SetUp.RateResolution |
| Segment.26.Ch1Time | Channel 1 time | time_t | 3f74 | 16244 | Set by Network.Modbus.TimeFormat |
| Segment.26.Ch1TSP | Channel 1 target set-point | float32 | 3f72 | 16242 | Same as Programmer.SetUp.Ch1PVInput |
| Segment.26.Ch1UserVal | Channel 1 user value | float32 | 3f88 | 16264 | Same as Programmer.SetUp.ResetCh1UserVal |
| Segment.26.Ch1Wait | Channel 1 Wait | uint8 | 3f7e | 16254 | Not applicable |
| Segment.26.Ch1WaitVal | Channel 1 wait value | float32 | 3f80 | 16256 | Same as Programmer.SetUp.PVWait1 |
| Segment.26.Ch2Holdback | Channel 2 holdback type | uint8 | 3f7a | 16250 | Not applicable |
| Segment.26.Ch2HoldbackVal | Channel 2 holdback value | float32 | 3f7c | 16252 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.26.Ch2PVEvent | Channel 2 PV event | uint8 | 3f85 | 16261 | Not applicable |
| Segment.26.Ch2PVEventUse | Channel 2 PV event use | bool | 3f93 | 16275 | Not applicable |
| Segment.26.Ch2PVEventVal | Channel 2 PV event value | float32 | 3f87 | 16263 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.26.Ch2Rate | Channel 2 rate | float32 | 3f77 | 16247 | Set by Programmer.SetUp.RateResolution |
| Segment.26.Ch2Time | Channel 2 time | time_t | 3f75 | 16245 | Set by Network.Modbus.TimeFormat |
| Segment.26.Ch2TSP | Channel 2 target set-point | float32 | 3f73 | 16243 | Same as Programmer.SetUp.Ch2PVInput |
| Segment.26.Ch2UserVal | Channel 2 user value | float32 | 3f89 | 16265 | Same as Programmer.SetUp.ResetCh2UserVal |
| Segment.26.Ch2Wait | Channel 2 Wait | uint8 | 3f7f | 16255 | Not applicable |
| Segment.26.Ch2WaitVal | Channel 2 wait value | float32 | 3f81 | 16257 | Same as Programmer.SetUp.PVWait2 |
| Segment.26.Cycles | Cycles | int16 | 3f83 | 16259 | Not applicable |
| Segment.26.Duration | Duration | time_t | 3f71 | 16241 | Set by Network.Modbus.TimeFormat |
| Segment.26.EndType | End type | uint8 | 3f78 | 16248 | Not applicable |
| Segment.26.Event1 | Event 1 | bool | 3f8a | 16266 | Not applicable |
| Segment.26.Event2 | Event 2 | bool | 3f8b | 16267 | Not applicable |
| Segment.26.Event3 | Event 3 | bool | 3f8c | 16268 | Not applicable |
| Segment.26.Event4 | Event 4 | bool | 3f8d | 16269 | Not applicable |
| Segment.26.Event5 | Event 5 | bool | 3f8e | 16270 | Not applicable |
| Segment.26.Event6 | Event 6 | bool | 3f8f | 16271 | Not applicable |
| Segment.26.Event7 | Event 7 | bool | 3f90 | 16272 | Not applicable |
| Segment.26.Event8 | Event 8 | bool | 3f91 | 16273 | Not applicable |
| Segment.26.GoBackTo | Go back to | uint8 | 3f82 | 16258 | Not applicable |
| Segment.26.SegmentName | Segment name | string_t | 6cdd | 27869 | Not applicable |
| Segment.26.Type | Type | uint8 | 3f70 | 16240 | Not applicable |
| Segment.26.WaitFor | Wait for | uint8 | 3f7d | 16253 | Not applicable |
| Steriliser.AutoCounter | Automatically increments the cycle number | bool | 2e0f | 11791 | Not applicable |
| Steriliser.CycleNumber | Current cycle number | int32 | 2e04 | 11780 | Not applicable |
| Steriliser.CycleStatus | The current cycle status.<br>0 = Waiting start 1 = Waiting 2 = Equilibrisation<br>3 = Sterilising 4 = Passed 5 = Failed<br>6 = Aborted 7 = Test cycle | uint8 | 2e08 | 11784 | Not applicable |
| Steriliser.CycleTime | The total cycle time | time_t | 2e25 | 11813 | Set by Network.Modbus.TimeFormat |
| Steriliser.EquilibrationTime | The equilibration time period for the current cycle. | time_t | 2e0c | 11788 | Set by Network.Modbus.TimeFormat |
| Steriliser.FailureDwell1 | Failure alarm dwell time for input 1 | time_t | 2e22 | 11810 | Set by Network.Modbus.TimeFormat |
| Steriliser.FailureDwell2 | Failure alarm dwell time for input 2 | time_t | 2e2b | 11819 | Set by Network.Modbus.TimeFormat |
| Steriliser.FailureDwell3 | Failure alarm dwell time for input 3 | time_t | 2e2c | 11820 | Set by Network.Modbus.TimeFormat |
| Steriliser.FailureDwell4 | Failure alarm dwell time for input 4 | time_t | 2e2d | 11821 | Set by Network.Modbus.TimeFormat |
| Steriliser.FileByTag | Name historical files by cycle number and tag<br>0 = File by Tag Off; 1 = File by Tag On | bool | 2e21 | 11809 | Not applicable |
| Steriliser.FileTag | Used as part of the historical filename | string_t | 68f7 | 26871 | Not applicable |
| Steriliser.Fvalue | F0 (A0) | time_t | 2e26 | 11814 | Set by Network.Modbus.TimeFormat |
| Steriliser.Input1PV | Input 1 | float32 | 2e00 | 11776 | 0dp |
| Steriliser.Input2PV | Input 2 | float32 | 2e01 | 11777 | 0dp |
| Steriliser.Input3PV | Input 3 | float32 | 2e02 | 11778 | 0dp |
| Steriliser.Input4PV | Input 4 | float32 | 2e03 | 11779 | 0dp |

For parameter values and settings (enumerations), see Segment 1

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| Steriliser.InputType1 | Input type 1<br>0 = Off 1 = thermocouple 2 = Rising pressure<br>3 = Falling pressure 4 = Rise air detect 5 = Fall air detect | uint8 | 2e1d | 11805 | Not applicable |
| Steriliser.InputType2 | Input type 2 (as Input type 1, above) | uint8 | 2e1e | 11806 | Not applicable |
| Steriliser.InputType3 | Input type 3 (as Input type 1, above) | uint8 | 2e1f | 11807 | Not applicable |
| Steriliser.InputType4 | Input type 4 (as Input type 1, above) | uint8 | 2e20 | 11808 | Not applicable |
| Steriliser.IP1BandHigh | Sterilisation temperature input 1 band high. | float32 | 2e0a | 11786 | Same as Steriliser.Input1PV |
| Steriliser.IP1BandLow | Sterilisation temperature input 1 band low. | float32 | 2e0b | 11787 | Same as Steriliser.Input1PV |
| Steriliser.IP1TargetSP | Input 1 target setpoint | float32 | 2e07 | 11783 | Same as Steriliser.Input1PV |
| Steriliser.IP2BandHigh | Sterilisation temperature input 2 band high. | float32 | 2e10 | 11792 | Same as Steriliser.Input2PV |
| Steriliser.IP2BandLow | Sterilisation temperature input 2 band low. | float32 | 2e11 | 11793 | Same as Steriliser.Input2PV |
| Steriliser.IP2TargetSP | Input 2 target setpoint | float32 | 2e16 | 11798 | Same as Steriliser.Input2PV |
| Steriliser.IP3BandHigh | Sterilisation temperature input 3 band high. | float32 | 2e12 | 11794 | Same as Steriliser.Input3PV |
| Steriliser.IP3BandLow | Sterilisation temperature input 3 band low. | float32 | 2e13 | 11795 | Same as Steriliser.Input3PV |
| Steriliser.IP3TargetSP | Input 3 target setpoint | float32 | 2e17 | 11799 | Same as Steriliser.Input3PV |
| Steriliser.IP4BandHigh | Sterilisation temperature input 4 band high. | float32 | 2e14 | 11796 | Same as Steriliser.Input4PV |
| Steriliser.IP4BandLow | Sterilisation temperature input 4 band low. | float32 | 2e15 | 11797 | Same as Steriliser.Input3PV |
| Steriliser.IP4TargetSP | Input 4 target setpoint | float32 | 2e18 | 11800 | Same as Steriliser.Input4PV |
| Steriliser.LowLimit | Low temperature limit for the F0 calculation. | float32 | 2e2a | 11818 | 0dp |
| Steriliser.MeasuredTemp | Measured Temperature used in the F0 calculation. | float32 | 2e27 | 11815 | 0dp |
| Steriliser.PassedOutput | 1 = cycle passed; 0 = cycle failed. | uint8 | 2e1c | 11804 | Not applicable |
| Steriliser.Remaining | The holding time remaining for the current cycle. | time_t | 2e0e | 11790 | Set by Network.Modbus.TimeFormat |
| Steriliser.RunningOutput | 1 = cycle running; 0 = cycle not running | uint8 | 2e1b | 11803 | Not applicable |
| Steriliser.Start121 | Start a predefined 121°C cycle | bool | 2e19 | 11801 | Not applicable |
| Steriliser.Start134 | Start a predefined 134°C cycle | bool | 2e1a | 11802 | Not applicable |
| Steriliser.StartCycle | Start a custom cycle | bool | 2e05 | 11781 | Not applicable |
| Steriliser.SterilisingTime | The total time the load was at sterilisation conditions. | time_t | 2e0d | 11789 | Set by Network.Modbus.TimeFormat |
| Steriliser.TargetTemperature | Target Temperature for the F0 calculation. | float32 | 2e29 | 11817 | 0dp |
| Steriliser.TargetTime | The target time of the sterilisation period. | time_t | 2e09 | 11785 | Set by Network.Modbus.TimeFormat |
| Steriliser.TargetTime121 | The target time for a 121°C cycle | time_t | 2e23 | 11811 | Set by Network.Modbus.TimeFormat |
| Steriliser.TargetTime134 | The target time for a 134°C cycle | time_t | 2e24 | 11812 | Set by Network.Modbus.TimeFormat |
| Steriliser.ZTemperatureInterval | The Z temperature interval for the F0 calculation. | float32 | 2e28 | 11816 | 0dp |
| Timer.1.ElapsedTime | Elapsed Time | time_t | 2ee0 | 12000 | Set by Network.Modbus.TimeFormat |
| Timer.1.In | Trigger/Gate input | bool | 2ee5 | 12005 | Not applicable |
| Timer.1.Out | Output (1 = On; 0 = Off) | bool | 2ee1 | 12001 | Not applicable |
| Timer.1.Time | Period for the timer (hh:mm:ss) | time_t | 2ee2 | 12002 | Set by Network.Modbus.TimeFormat |
| Timer.1.Triggered | 1 = Timer triggered; 0 = Timer not triggered | bool | 2ee3 | 12003 | Not applicable |
| Timer.1.Type | Type of Timer<br>0 = Disabled (off) 1 = On Pulse 2 = On delay<br>3 = One shot 4 = Min on. | uint8 | 2ee4 | 12004 | Not applicable |
| Timer.2.ElapsedTime | Elapsed Time | time_t | 2ee6 | 12006 | Set by Network.Modbus.TimeFormat |
| Timer.2.In | Trigger/Gate input | bool | 2eeb | 12011 | Not applicable |
| Timer.2.Out | Output (1 = On; 0 = Off) | bool | 2ee7 | 12007 | Not applicable |
| Timer.2.Time | Period for the timer (hh:mm:ss) | time_t | 2ee8 | 12008 | Set by Network.Modbus.TimeFormat |
| Timer.2.Triggered | 1 = Timer triggered; 0 = Timer not triggered | bool | 2ee9 | 12009 | Not applicable |
| Timer.2.Type | Type of Timer (as Timer.1.Type) | uint8 | 2eea | 12010 | Not applicable |
| Timer.3.ElapsedTime | Elapsed Time | time_t | 2eec | 12012 | Set by Network.Modbus.TimeFormat |
| Timer.3.In | Trigger/Gate input | bool | 2ef1 | 12017 | Not applicable |
| Timer.3.Out | Output (1 = On; 0 = Off) | bool | 2eed | 12013 | Not applicable |
| Timer.3.Time | Period for the timer (hh:mm:ss) | time_t | 2eee | 12014 | Set by Network.Modbus.TimeFormat |
| Timer.3.Triggered | 1 = Timer triggered; 0 = Timer not triggered | bool | 2eef | 12015 | Not applicable |
| Timer.3.Type | Type of Timer (as Timer.1.Type) | uint8 | 2ef0 | 12016 | Not applicable |
| Timer.4.ElapsedTime | Elapsed Time | time_t | 2ef2 | 12018 | Set by Network.Modbus.TimeFormat |
| Timer.4.In | Trigger/Gate input | bool | 2ef7 | 12023 | Not applicable |
| Timer.4.Out | Output (1 = On; 0 = Off) | bool | 2ef3 | 12019 | Not applicable |
| Timer.4.Time | Period for the timer (hh:mm:ss) | time_t | 2ef4 | 12020 | Set by Network.Modbus.TimeFormat |
| Timer.4.Triggered | 1 = Timer triggered; 0 = Timer not triggered | bool | 2ef5 | 12021 | Not applicable |
| Timer.4.Type | Type of Timer (as Timer.1.Type) | uint8 | 2ef6 | 12022 | Not applicable |
| UserLin.1.NumberOfBreakpoints | Number of points in user linearisation table 1 | uint8 | 2900 | 10496 | Not applicable |
| UserLin.1.X1 | User linearisation table 1 ‘X’ value 1 | float32 | 2901 | 10497 | 2dp |
| UserLin.1.X2 | User linearisation table 1 ‘X’ value 2 | float32 | 2903 | 10499 | 2dp |
| UserLin.1.X3 | User linearisation table 1 ‘X’ value 3 | float32 | 2905 | 10501 | 2dp |
| UserLin.1.X4 | User linearisation table 1 ‘X’ value 4 | float32 | 2907 | 10503 | 2dp |
| UserLin.1.X5 | User linearisation table 1 ‘X’ value 5 | float32 | 2909 | 10505 | 2dp |
| UserLin.1.X6 | User linearisation table 1 ‘X’ value 6 | float32 | 290b | 10507 | 2dp |
| UserLin.1.X7 | User linearisation table 1 ‘X’ value 7 | float32 | 290d | 10509 | 2dp |
| UserLin.1.X8 | User linearisation table 1 ‘X’ value 8 | float32 | 290f | 10511 | 2dp |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| UserLin.1.X9 | User linearisation table 1 ‘X’ value 9 | float32 | 2911 | 10513 | 2dp |
| UserLin.1.X10 | User linearisation table 1 ‘X’ value 10 | float32 | 2913 | 10515 | 2dp |
| UserLin.1.X11 | User linearisation table 1 ‘X’ value 11 | float32 | 2915 | 10517 | 2dp |
| UserLin.1.X12 | User linearisation table 1 ‘X’ value 12 | float32 | 2917 | 10519 | 2dp |
| UserLin.1.X13 | User linearisation table 1 ‘X’ value 13 | float32 | 2919 | 10521 | 2dp |
| UserLin.1.X14 | User linearisation table 1 ‘X’ value 14 | float32 | 291b | 10523 | 2dp |
| UserLin.1.X15 | User linearisation table 1 ‘X’ value 15 | float32 | 291d | 10525 | 2dp |
| UserLin.1.X16 | User linearisation table 1 ‘X’ value 16 | float32 | 291f | 10527 | 2dp |
| UserLin.1.X17 | User linearisation table 1 ‘X’ value 17 | float32 | 2921 | 10529 | 2dp |
| UserLin.1.X18 | User linearisation table 1 ‘X’ value 18 | float32 | 2923 | 10531 | 2dp |
| UserLin.1.X19 | User linearisation table 1 ‘X’ value 19 | float32 | 2925 | 10533 | 2dp |
| UserLin.1.X20 | User linearisation table 1 ‘X’ value 20 | float32 | 2927 | 10535 | 2dp |
| UserLin.1.X21 | User linearisation table 1 ‘X’ value 21 | float32 | 2929 | 10537 | 2dp |
| UserLin.1.X22 | User linearisation table 1 ‘X’ value 22 | float32 | 292b | 10539 | 2dp |
| UserLin.1.X23 | User linearisation table 1 ‘X’ value 23 | float32 | 292d | 10541 | 2dp |
| UserLin.1.X24 | User linearisation table 1 ‘X’ value 24 | float32 | 292f | 10543 | 2dp |
| UserLin.1.X25 | User linearisation table 1 ‘X’ value 25 | float32 | 2931 | 10545 | 2dp |
| UserLin.1.X26 | User linearisation table 1 ‘X’ value 26 | float32 | 2933 | 10547 | 2dp |
| UserLin.1.X27 | User linearisation table 1 ‘X’ value 27 | float32 | 2935 | 10549 | 2dp |
| UserLin.1.X28 | User linearisation table 1 ‘X’ value 28 | float32 | 2937 | 10551 | 2dp |
| UserLin.1.X29 | User linearisation table 1 ‘X’ value 29 | float32 | 2939 | 10553 | 2dp |
| UserLin.1.X30 | User linearisation table 1 ‘X’ value 30 | float32 | 293b | 10555 | 2dp |
| UserLin.1.X31 | User linearisation table 1 ‘X’ value 31 | float32 | 293d | 10557 | 2dp |
| UserLin.1.X32 | User linearisation table 1 ‘X’ value 32 | float32 | 293f | 10559 | 2dp |
| UserLin.1.Y1 | User linearisation table 1 ‘Y’ value 1 | float32 | 2902 | 10498 | 2dp |
| UserLin.1.Y2 | User linearisation table 1 ‘Y’ value 2 | float32 | 2904 | 10500 | 2dp |
| UserLin.1.Y3 | User linearisation table 1 ‘Y’ value 3 | float32 | 2906 | 10502 | 2dp |
| UserLin.1.Y4 | User linearisation table 1 ‘Y’ value 4 | float32 | 2908 | 10504 | 2dp |
| UserLin.1.Y5 | User linearisation table 1 ‘Y’ value 5 | float32 | 290a | 10506 | 2dp |
| UserLin.1.Y6 | User linearisation table 1 ‘Y’ value 6 | float32 | 290c | 10508 | 2dp |
| UserLin.1.Y7 | User linearisation table 1 ‘Y’ value 7 | float32 | 290e | 10510 | 2dp |
| UserLin.1.Y8 | User linearisation table 1 ‘Y’ value 8 | float32 | 2910 | 10512 | 2dp |
| UserLin.1.Y9 | User linearisation table 1 ‘Y’ value 9 | float32 | 2912 | 10514 | 2dp |
| UserLin.1.Y10 | User linearisation table 1 ‘Y’ value 10 | float32 | 2914 | 10516 | 2dp |
| UserLin.1.Y11 | User linearisation table 1 ‘Y’ value 11 | float32 | 2916 | 10518 | 2dp |
| UserLin.1.Y12 | User linearisation table 1 ‘Y’ value 12 | float32 | 2918 | 10520 | 2dp |
| UserLin.1.Y13 | User linearisation table 1 ‘Y’ value 13 | float32 | 291a | 10522 | 2dp |
| UserLin.1.Y14 | User linearisation table 1 ‘Y’ value 14 | float32 | 291c | 10524 | 2dp |
| UserLin.1.Y15 | User linearisation table 1 ‘Y’ value 15 | float32 | 291e | 10526 | 2dp |
| UserLin.1.Y16 | User linearisation table 1 ‘Y’ value 16 | float32 | 2920 | 10528 | 2dp |
| UserLin.1.Y17 | User linearisation table 1 ‘Y’ value 17 | float32 | 2922 | 10530 | 2dp |
| UserLin.1.Y18 | User linearisation table 1 ‘Y’ value 18 | float32 | 2924 | 10532 | 2dp |
| UserLin.1.Y19 | User linearisation table 1 ‘Y’ value 19 | float32 | 2926 | 10534 | 2dp |
| UserLin.1.Y20 | User linearisation table 1 ‘Y’ value 20 | float32 | 2928 | 10536 | 2dp |
| UserLin.1.Y21 | User linearisation table 1 ‘Y’ value 21 | float32 | 292a | 10538 | 2dp |
| UserLin.1.Y22 | User linearisation table 1 ‘Y’ value 22 | float32 | 292c | 10540 | 2dp |
| UserLin.1.Y23 | User linearisation table 1 ‘Y’ value 23 | float32 | 292e | 10542 | 2dp |
| UserLin.1.Y24 | User linearisation table 1 ‘Y’ value 24 | float32 | 2930 | 10544 | 2dp |
| UserLin.1.Y25 | User linearisation table 1 ‘Y’ value 25 | float32 | 2932 | 10546 | 2dp |
| UserLin.1.Y26 | User linearisation table 1 ‘Y’ value 26 | float32 | 2934 | 10548 | 2dp |
| UserLin.1.Y27 | User linearisation table 1 ‘Y’ value 27 | float32 | 2936 | 10550 | 2dp |
| UserLin.1.Y28 | User linearisation table 1 ‘Y’ value 28 | float32 | 2938 | 10552 | 2dp |
| UserLin.1.Y29 | User linearisation table 1 ‘Y’ value 29 | float32 | 293a | 10554 | 2dp |
| UserLin.1.Y30 | User linearisation table 1 ‘Y’ value 30 | float32 | 293c | 10556 | 2dp |
| UserLin.1.Y31 | User linearisation table 1 ‘Y’ value 31 | float32 | 293e | 10558 | 2dp |
| UserLin.1.Y32 | User linearisation table 1 ‘Y’ value 32 | float32 | 2940 | 10560 | 2dp |
| UserLin.2.NumberOfBreakpoints | Number of points in user linearisation table 2 | uint8 | 29c0 | 10688 | Not applicable |
| UserLin.2.X1 | User linearisation table 2 ‘X’ value 1 | float32 | 29c1 | 10689 | 2dp |
| UserLin.2.X2 | User linearisation table 2 ‘X’ value 2 | float32 | 29c3 | 10691 | 2dp |
| UserLin.2.X3 | User linearisation table 2 ‘X’ value 3 | float32 | 29c5 | 10693 | 2dp |
| UserLin.2.X4 | User linearisation table 2 ‘X’ value 4 | float32 | 29c7 | 10695 | 2dp |
| UserLin.2.X5 | User linearisation table 2 ‘X’ value 5 | float32 | 29c9 | 10697 | 2dp |
| UserLin.2.X6 | User linearisation table 2 ‘X’ value 6 | float32 | 29cb | 10699 | 2dp |
| UserLin.2.X7 | User linearisation table 2 ‘X’ value 7 | float32 | 29cd | 10701 | 2dp |
| UserLin.2.X8 | User linearisation table 2 ‘X’ value 8 | float32 | 29cf | 10703 | 2dp |
| UserLin.2.X9 | User linearisation table 2 ‘X’ value 9 | float32 | 29d1 | 10705 | 2dp |
| UserLin.2.X10 | User linearisation table 2 ‘X’ value 10 | float32 | 29d3 | 10707 | 2dp |
| UserLin.2.X11 | User linearisation table 2 ‘X’ value 11 | float32 | 29d5 | 10709 | 2dp |
| UserLin.2.X12 | User linearisation table 2 ‘X’ value 12 | float32 | 29d7 | 10711 | 2dp |
| UserLin.2.X13 | User linearisation table 2 ‘X’ value 13 | float32 | 29d9 | 10713 | 2dp |
| UserLin.2.X14 | User linearisation table 2 ‘X’ value 14 | float32 | 29db | 10715 | 2dp |
| UserLin.2.X15 | User linearisation table 2 ‘X’ value 15 | float32 | 29dd | 10717 | 2dp |
| UserLin.2.X16 | User linearisation table 2 ‘X’ value 16 | float32 | 29df | 10719 | 2dp |
| UserLin.2.X17 | User linearisation table 2 ‘X’ value 17 | float32 | 29e1 | 10721 | 2dp |
| UserLin.2.X18 | User linearisation table 2 ‘X’ value 18 | float32 | 29e3 | 10723 | 2dp |
| UserLin.2.X19 | User linearisation table 2 ‘X’ value 19 | float32 | 29e5 | 10725 | 2dp |
| UserLin.2.X20 | User linearisation table 2 ‘X’ value 20 | float32 | 29e7 | 10727 | 2dp |
| UserLin.2.X21 | User linearisation table 2 ‘X’ value 21 | float32 | 29e9 | 10729 | 2dp |
| UserLin.2.X22 | User linearisation table 2 ‘X’ value 22 | float32 | 29eb | 10731 | 2dp |
| UserLin.2.X23 | User linearisation table 2 ‘X’ value 23 | float32 | 29ed | 10733 | 2dp |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| UserLin.2.X24 | User linearisation table 2 ‘X’ value 24 | float32 | 29ef | 10735 | 2dp |
| UserLin.2.X25 | User linearisation table 2 ‘X’ value 25 | float32 | 29f1 | 10737 | 2dp |
| UserLin.2.X26 | User linearisation table 2 ‘X’ value 26 | float32 | 29f3 | 10739 | 2dp |
| UserLin.2.X27 | User linearisation table 2 ‘X’ value 27 | float32 | 29f5 | 10741 | 2dp |
| UserLin.2.X28 | User linearisation table 2 ‘X’ value 28 | float32 | 29f7 | 10743 | 2dp |
| UserLin.2.X29 | User linearisation table 2 ‘X’ value 29 | float32 | 29f9 | 10745 | 2dp |
| UserLin.2.X30 | User linearisation table 2 ‘X’ value 30 | float32 | 29fb | 10747 | 2dp |
| UserLin.2.X31 | User linearisation table 2 ‘X’ value 31 | float32 | 29fd | 10749 | 2dp |
| UserLin.2.X32 | User linearisation table 2 ‘X’ value 32 | float32 | 29ff | 10751 | 2dp |
| UserLin.2.Y1 | User linearisation table 2 ‘Y’ value 1 | float32 | 29c2 | 10690 | 2dp |
| UserLin.2.Y2 | User linearisation table 4 ‘Y’ value 2 | float32 | 29c4 | 10692 | 2dp |
| UserLin.2.Y3 | User linearisation table 4 ‘Y’ value 3 | float32 | 29c6 | 10694 | 2dp |
| UserLin.2.Y4 | User linearisation table 4 ‘Y’ value 4 | float32 | 29c8 | 10696 | 2dp |
| UserLin.2.Y5 | User linearisation table 4 ‘Y’ value 5 | float32 | 29ca | 10698 | 2dp |
| UserLin.2.Y6 | User linearisation table 4 ‘Y’ value 6 | float32 | 29cc | 10700 | 2dp |
| UserLin.2.Y7 | User linearisation table 4 ‘Y’ value 7 | float32 | 29ce | 10702 | 2dp |
| UserLin.2.Y8 | User linearisation table 4 ‘Y’ value 8 | float32 | 29d0 | 10704 | 2dp |
| UserLin.2.Y9 | User linearisation table 4 ‘Y’ value 9 | float32 | 29d2 | 10706 | 2dp |
| UserLin.2.Y10 | User linearisation table 4 ‘Y’ value 10 | float32 | 29d4 | 10708 | 2dp |
| UserLin.2.Y11 | User linearisation table 4 ‘Y’ value 11 | float32 | 29d6 | 10710 | 2dp |
| UserLin.2.Y12 | User linearisation table 4 ‘Y’ value 12 | float32 | 29d8 | 10712 | 2dp |
| UserLin.2.Y13 | User linearisation table 4 ‘Y’ value 13 | float32 | 29da | 10714 | 2dp |
| UserLin.2.Y14 | User linearisation table 4 ‘Y’ value 14 | float32 | 29dc | 10716 | 2dp |
| UserLin.2.Y15 | User linearisation table 4 ‘Y’ value 15 | float32 | 29de | 10718 | 2dp |
| UserLin.2.Y16 | User linearisation table 4 ‘Y’ value 16 | float32 | 29e0 | 10720 | 2dp |
| UserLin.2.Y17 | User linearisation table 4 ‘Y’ value 17 | float32 | 29e2 | 10722 | 2dp |
| UserLin.2.Y18 | User linearisation table 4 ‘Y’ value 18 | float32 | 29e4 | 10724 | 2dp |
| UserLin.2.Y19 | User linearisation table 4 ‘Y’ value 19 | float32 | 29e6 | 10726 | 2dp |
| UserLin.2.Y20 | User linearisation table 4 ‘Y’ value 20 | float32 | 29e8 | 10728 | 2dp |
| UserLin.2.Y21 | User linearisation table 4 ‘Y’ value 21 | float32 | 29ea | 10730 | 2dp |
| UserLin.2.Y22 | User linearisation table 4 ‘Y’ value 22 | float32 | 29ec | 10732 | 2dp |
| UserLin.2.Y23 | User linearisation table 4 ‘Y’ value 23 | float32 | 29ee | 10734 | 2dp |
| UserLin.2.Y24 | User linearisation table 4 ‘Y’ value 24 | float32 | 29f0 | 10736 | 2dp |
| UserLin.2.Y25 | User linearisation table 4 ‘Y’ value 25 | float32 | 29f2 | 10738 | 2dp |
| UserLin.2.Y26 | User linearisation table 4 ‘Y’ value 26 | float32 | 29f4 | 10740 | 2dp |
| UserLin.2.Y27 | User linearisation table 4 ‘Y’ value 27 | float32 | 29f6 | 10742 | 2dp |
| UserLin.2.Y28 | User linearisation table 4 ‘Y’ value 28 | float32 | 29f8 | 10744 | 2dp |
| UserLin.2.Y29 | User linearisation table 4 ‘Y’ value 29 | float32 | 29fa | 10746 | 2dp |
| UserLin.2.Y30 | User linearisation table 4 ‘Y’ value 30 | float32 | 29fc | 10748 | 2dp |
| UserLin.2.Y31 | User linearisation table 4 ‘Y’ value 31 | float32 | 29fe | 10750 | 2dp |
| UserLin.2.Y32 | User linearisation table 4 ‘Y’ value 32 | float32 | 2a00 | 10752 | 2dp |
| | | | | | |
| UserLin.3.NumberOfBreakpoints | Number of points in user linearisation table 32 | uint8 | 2a80 | 10880 | Not applicable |
| UserLin.3.X1 | User linearisation table 3 ‘X’ value 1 | float32 | 2a81 | 10881 | 2dp |
| UserLin.3.X2 | User linearisation table 3 ‘X’ value 2 | float32 | 2a83 | 10883 | 2dp |
| UserLin.3.X3 | User linearisation table 3 ‘X’ value 3 | float32 | 2a85 | 10885 | 2dp |
| UserLin.3.X4 | User linearisation table 3 ‘X’ value 4 | float32 | 2a87 | 10887 | 2dp |
| UserLin.3.X5 | User linearisation table 3 ‘X’ value 5 | float32 | 2a89 | 10889 | 2dp |
| UserLin.3.X6 | User linearisation table 3 ‘X’ value 6 | float32 | 2a8b | 10891 | 2dp |
| UserLin.3.X7 | User linearisation table 3 ‘X’ value 7 | float32 | 2a8d | 10893 | 2dp |
| UserLin.3.X8 | User linearisation table 3 ‘X’ value 8 | float32 | 2a8f | 10895 | 2dp |
| UserLin.3.X9 | User linearisation table 3 ‘X’ value 9 | float32 | 2a91 | 10897 | 2dp |
| UserLin.3.X10 | User linearisation table 3 ‘X’ value 10 | float32 | 2a93 | 10899 | 2dp |
| UserLin.3.X11 | User linearisation table 3 ‘X’ value 11 | float32 | 2a95 | 10901 | 2dp |
| UserLin.3.X12 | User linearisation table 3 ‘X’ value 12 | float32 | 2a97 | 10903 | 2dp |
| UserLin.3.X13 | User linearisation table 3 ‘X’ value 13 | float32 | 2a99 | 10905 | 2dp |
| UserLin.3.X14 | User linearisation table 3 ‘X’ value 14 | float32 | 2a9b | 10907 | 2dp |
| UserLin.3.X15 | User linearisation table 3 ‘X’ value 15 | float32 | 2a9d | 10909 | 2dp |
| UserLin.3.X16 | User linearisation table 3 ‘X’ value 16 | float32 | 2a9f | 10911 | 2dp |
| UserLin.3.X17 | User linearisation table 3 ‘X’ value 17 | float32 | 2aa1 | 10913 | 2dp |
| UserLin.3.X18 | User linearisation table 3 ‘X’ value 18 | float32 | 2aa3 | 10915 | 2dp |
| UserLin.3.X19 | User linearisation table 3 ‘X’ value 19 | float32 | 2aa5 | 10917 | 2dp |
| UserLin.3.X20 | User linearisation table 3 ‘X’ value 20 | float32 | 2aa7 | 10919 | 2dp |
| UserLin.3.X21 | User linearisation table 3 ‘X’ value 21 | float32 | 2aa9 | 10921 | 2dp |
| UserLin.3.X22 | User linearisation table 3 ‘X’ value 22 | float32 | 2aab | 10923 | 2dp |
| UserLin.3.X23 | User linearisation table 3 ‘X’ value 23 | float32 | 2aad | 10925 | 2dp |
| UserLin.3.X24 | User linearisation table 3 ‘X’ value 24 | float32 | 2aaf | 10927 | 2dp |
| UserLin.3.X25 | User linearisation table 3 ‘X’ value 25 | float32 | 2ab1 | 10929 | 2dp |
| UserLin.3.X26 | User linearisation table 3 ‘X’ value 26 | float32 | 2ab3 | 10931 | 2dp |
| UserLin.3.X27 | User linearisation table 3 ‘X’ value 27 | float32 | 2ab5 | 10933 | 2dp |
| UserLin.3.X28 | User linearisation table 3 ‘X’ value 28 | float32 | 2ab7 | 10935 | 2dp |
| UserLin.3.X29 | User linearisation table 3 ‘X’ value 29 | float32 | 2ab9 | 10937 | 2dp |
| UserLin.3.X30 | User linearisation table 3 ‘X’ value 30 | float32 | 2abb | 10939 | 2dp |
| UserLin.3.X31 | User linearisation table 3 ‘X’ value 31 | float32 | 2abd | 10941 | 2dp |
| UserLin.3.X32 | User linearisation table 3 ‘X’ value 32 | float32 | 2abf | 10943 | 2dp |
| UserLin.3.Y1 | User linearisation table 4 ‘Y’ value 1 | float32 | 2a82 | 10882 | 2dp |
| UserLin.3.Y2 | User linearisation table 4 ‘Y’ value 2 | float32 | 2a84 | 10884 | 2dp |
| UserLin.3.Y3 | User linearisation table 4 ‘Y’ value 3 | float32 | 2a86 | 10886 | 2dp |
| UserLin.3.Y4 | User linearisation table 4 ‘Y’ value 4 | float32 | 2a88 | 10888 | 2dp |
| UserLin.3.Y5 | User linearisation table 4 ‘Y’ value 5 | float32 | 2a8a | 10890 | 2dp |
| UserLin.3.Y6 | User linearisation table 4 ‘Y’ value 6 | float32 | 2a8c | 10892 | 2dp |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| UserLin.3.Y7 | User linearisation table 4 'Y' value 7 | float32 | 2a8e | 10894 | 2dp |
| UserLin.3.Y8 | User linearisation table 4 'Y' value 8 | float32 | 2a90 | 10896 | 2dp |
| UserLin.3.Y9 | User linearisation table 4 'Y' value 9 | float32 | 2a92 | 10898 | 2dp |
| UserLin.3.Y10 | User linearisation table 4 'Y' value 10 | float32 | 2a94 | 10900 | 2dp |
| UserLin.3.Y11 | User linearisation table 4 'Y' value 11 | float32 | 2a96 | 10902 | 2dp |
| UserLin.3.Y12 | User linearisation table 4 'Y' value 12 | float32 | 2a98 | 10904 | 2dp |
| UserLin.3.Y13 | User linearisation table 4 'Y' value 13 | float32 | 2a9a | 10906 | 2dp |
| UserLin.3.Y14 | User linearisation table 4 'Y' value 14 | float32 | 2a9c | 10908 | 2dp |
| UserLin.3.Y15 | User linearisation table 4 'Y' value 15 | float32 | 2a9e | 10910 | 2dp |
| UserLin.3.Y16 | User linearisation table 4 'Y' value 16 | float32 | 2aa0 | 10912 | 2dp |
| UserLin.3.Y17 | User linearisation table 4 'Y' value 17 | float32 | 2aa2 | 10914 | 2dp |
| UserLin.3.Y18 | User linearisation table 4 'Y' value 18 | float32 | 2aa4 | 10916 | 2dp |
| UserLin.3.Y19 | User linearisation table 4 'Y' value 19 | float32 | 2aa6 | 10918 | 2dp |
| UserLin.3.Y20 | User linearisation table 4 'Y' value 20 | float32 | 2aa8 | 10920 | 2dp |
| UserLin.3.Y21 | User linearisation table 4 'Y' value 21 | float32 | 2aaa | 10922 | 2dp |
| UserLin.3.Y22 | User linearisation table 4 'Y' value 22 | float32 | 2aac | 10924 | 2dp |
| UserLin.3.Y23 | User linearisation table 4 'Y' value 23 | float32 | 2aae | 10926 | 2dp |
| UserLin.3.Y24 | User linearisation table 4 'Y' value 24 | float32 | 2ab0 | 10928 | 2dp |
| UserLin.3.Y25 | User linearisation table 4 'Y' value 25 | float32 | 2ab2 | 10930 | 2dp |
| UserLin.3.Y26 | User linearisation table 4 'Y' value 26 | float32 | 2ab4 | 10932 | 2dp |
| UserLin.3.Y27 | User linearisation table 4 'Y' value 27 | float32 | 2ab6 | 10934 | 2dp |
| UserLin.3.Y28 | User linearisation table 4 'Y' value 28 | float32 | 2ab8 | 10936 | 2dp |
| UserLin.3.Y29 | User linearisation table 4 'Y' value 29 | float32 | 2aba | 10938 | 2dp |
| UserLin.3.Y30 | User linearisation table 4 'Y' value 30 | float32 | 2abc | 10940 | 2dp |
| UserLin.3.Y31 | User linearisation table 4 'Y' value 31 | float32 | 2abe | 10942 | 2dp |
| UserLin.3.Y32 | User linearisation table 4 'Y' value 32 | float32 | 2ac0 | 10944 | 2dp |
| UserLin.4.NumberOfBreakpoints | Number of points in user linearisation table 4 | uint8 | 2b40 | 11072 | Not applicable |
| UserLin.4.X1 | User linearisation table 4 'X' value 1 | float32 | 2b41 | 11073 | 2dp |
| UserLin.4.X2 | User linearisation table 4 'X' value 2 | float32 | 2b43 | 11075 | 2dp |
| UserLin.4.X3 | User linearisation table 4 'X' value 3 | float32 | 2b45 | 11077 | 2dp |
| UserLin.4.X4 | User linearisation table 4 'X' value 4 | float32 | 2b47 | 11079 | 2dp |
| UserLin.4.X5 | User linearisation table 4 'X' value v5 | float32 | 2b49 | 11081 | 2dp |
| UserLin.4.X6 | User linearisation table 4 'X' value 6 | float32 | 2b4b | 11083 | 2dp |
| UserLin.4.X7 | User linearisation table 4 'X' value 7 | float32 | 2b4d | 11085 | 2dp |
| UserLin.4.X8 | User linearisation table 4 'X' value 8 | float32 | 2b4f | 11087 | 2dp |
| UserLin.4.X9 | User linearisation table 4 'X' value 9 | float32 | 2b51 | 11089 | 2dp |
| UserLin.4.X10 | User linearisation table 4 'X' value 10 | float32 | 2b53 | 11091 | 2dp |
| UserLin.4.X11 | User linearisation table 4 'X' value 11 | float32 | 2b55 | 11093 | 2dp |
| UserLin.4.X12 | User linearisation table 4 'X' value 12 | float32 | 2b57 | 11095 | 2dp |
| UserLin.4.X13 | User linearisation table 4 'X' value 13 | float32 | 2b59 | 11097 | 2dp |
| UserLin.4.X14 | User linearisation table 4 'X' value 14 | float32 | 2b5b | 11099 | 2dp |
| UserLin.4.X15 | User linearisation table 4 'X' value 15 | float32 | 2b5d | 11101 | 2dp |
| UserLin.4.X16 | User linearisation table 4 'X' value 16 | float32 | 2b5f | 11103 | 2dp |
| UserLin.4.X17 | User linearisation table 4 'X' value 17 | float32 | 2b61 | 11105 | 2dp |
| UserLin.4.X18 | User linearisation table 4 'X' value 18 | float32 | 2b63 | 11107 | 2dp |
| UserLin.4.X19 | User linearisation table 4 'X' value 19 | float32 | 2b65 | 11109 | 2dp |
| UserLin.4.X20 | User linearisation table 4 'X' value 20 | float32 | 2b67 | 11111 | 2dp |
| UserLin.4.X21 | User linearisation table 4 'X' value 21 | float32 | 2b69 | 11113 | 2dp |
| UserLin.4.X22 | User linearisation table 4 'X' value 22 | float32 | 2b6b | 11115 | 2dp |
| UserLin.4.X23 | User linearisation table 4 'X' value 23 | float32 | 2b6d | 11117 | 2dp |
| UserLin.4.X24 | User linearisation table 4 'X' value 24 | float32 | 2b6f | 11119 | 2dp |
| UserLin.4.X25 | User linearisation table 4 'X' value 25 | float32 | 2b71 | 11121 | 2dp |
| UserLin.4.X26 | User linearisation table 4 'X' value 26 | float32 | 2b73 | 11123 | 2dp |
| UserLin.4.X27 | User linearisation table 4 'X' value 27 | float32 | 2b75 | 11125 | 2dp |
| UserLin.4.X28 | User linearisation table 4 'X' value 28 | float32 | 2b77 | 11127 | 2dp |
| UserLin.4.X29 | User linearisation table 4 'X' value 29 | float32 | 2b79 | 11129 | 2dp |
| UserLin.4.X30 | User linearisation table 4 'X' value 30 | float32 | 2b7b | 11131 | 2dp |
| UserLin.4.X31 | User linearisation table 4 'X' value 31 | float32 | 2b7d | 11133 | 2dp |
| UserLin.4.X32 | User linearisation table 4 'X' value 32 | float32 | 2b7f | 11135 | 2dp |
| UserLin.4.Y1 | User linearisation table 4 'Y' value 1 | float32 | 2b42 | 11074 | 2dp |
| UserLin.4.Y2 | User linearisation table 4 'Y' value 2 | float32 | 2b44 | 11076 | 2dp |
| UserLin.4.Y3 | User linearisation table 4 'Y' value 3 | float32 | 2b46 | 11078 | 2dp |
| UserLin.4.Y4 | User linearisation table 4 'Y' value 4 | float32 | 2b48 | 11080 | 2dp |
| UserLin.4.Y5 | User linearisation table 4 'Y' value 5 | float32 | 2b4a | 11082 | 2dp |
| UserLin.4.Y6 | User linearisation table 4 'Y' value 6 | float32 | 2b4c | 11084 | 2dp |
| UserLin.4.Y7 | User linearisation table 4 'Y' value 7 | float32 | 2b4e | 11086 | 2dp |
| UserLin.4.Y8 | User linearisation table 4 'Y' value 8 | float32 | 2b50 | 11088 | 2dp |
| UserLin.4.Y9 | User linearisation table 4 'Y' value 9 | float32 | 2b52 | 11090 | 2dp |
| UserLin.4.Y10 | User linearisation table 4 'Y' value 10 | float32 | 2b54 | 11092 | 2dp |
| UserLin.4.Y11 | User linearisation table 4 'Y' value 11 | float32 | 2b56 | 11094 | 2dp |
| UserLin.4.Y12 | User linearisation table 4 'Y' value 12 | float32 | 2b58 | 11096 | 2dp |
| UserLin.4.Y13 | User linearisation table 4 'Y' value 13 | float32 | 2b5a | 11098 | 2dp |
| UserLin.4.Y14 | User linearisation table 4 'Y' value 14 | float32 | 2b5c | 11100 | 2dp |
| UserLin.4.Y15 | User linearisation table 4 'Y' value 15 | float32 | 2b5e | 11102 | 2dp |
| UserLin.4.Y16 | User linearisation table 4 'Y' value 16 | float32 | 2b60 | 11104 | 2dp |
| UserLin.4.Y17 | User linearisation table 4 'Y' value 17 | float32 | 2b62 | 11106 | 2dp |
| UserLin.4.Y18 | User linearisation table 4 'Y' value 18 | float32 | 2b64 | 11108 | 2dp |
| UserLin.4.Y19 | User linearisation table 4 'Y' value 19 | float32 | 2b66 | 11110 | 2dp |
| UserLin.4.Y20 | User linearisation table 4 'Y' value 20 | float32 | 2b68 | 11112 | 2dp |
| UserLin.4.Y21 | User linearisation table 4 'Y' value 21 | float32 | 2b6a | 11114 | 2dp |
| UserLin.4.Y22 | User linearisation table 4 'Y' value 22 | float32 | 2b6c | 11116 | 2dp |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| UserLin.4.Y23 | User linearisation table 4 'Y' value 23 | float32 | 2b6e | 11118 | 2dp |
| UserLin.4.Y24 | User linearisation table 4 'Y' value 24 | float32 | 2b70 | 11120 | 2dp |
| UserLin.4.Y25 | User linearisation table 4 'Y' value 25 | float32 | 2b72 | 11122 | 2dp |
| UserLin.4.Y26 | User linearisation table 4 'Y' value 26 | float32 | 2b74 | 11124 | 2dp |
| UserLin.4.Y27 | User linearisation table 4 'Y' value 27 | float32 | 2b76 | 11126 | 2dp |
| UserLin.4.Y28 | User linearisation table 4 'Y' value 28 | float32 | 2b78 | 11128 | 2dp |
| UserLin.4.Y29 | User linearisation table 4 'Y' value 29 | float32 | 2b7a | 11130 | 2dp |
| UserLin.4.Y30 | User linearisation table 4 'Y' value 30 | float32 | 2b7c | 11132 | 2dp |
| UserLin.4.Y31 | User linearisation table 4 'Y' value 31 | float32 | 2b7e | 11134 | 2dp |
| UserLin.4.Y32 | User linearisation table 4 'Y' value 32 | float32 | 2b80 | 11136 | 2dp |
| UsrVal.1.HighLimit | User Value High Limit | float32 | 2e8c | 11916 | Set by UsrVal.1.Resolution |
| UsrVal.1.LowLimit | User Value Low Limit | float32 | 2e8d | 11917 | Set by UsrVal.1.Resolution |
| UsrVal.1.Resolution | Result Resolution | uint8 | 2e90 | 11920 | Not applicable |
| UsrVal.1.Status | User Value 1 Status (0 = Good (OK); 7 = Bad (Error)) | bool | 2e8f | 11919 | Not applicable |
| UsrVal.1.Units | Units of the value | string_t | 68fc | 26876 | Not applicable |
| UsrVal.1.Val | The User Value | float32 | 2e8e | 11918 | Set by UsrVal.1.Resolution |
| UsrVal.2.HighLimit | User Value High Limit | float32 | 2e91 | 11921 | Set by UsrVal.2.Resolution |
| UsrVal.2.LowLimit | User Value Low Limit | float32 | 2e92 | 11922 | Set by UsrVal.2.Resolution |
| UsrVal.2.Resolution | Result Resolution | uint8 | 2e95 | 11925 | Not applicable |
| UsrVal.2.Status | User Value 2 Status (0 = Good (OK); 7 = Bad (Error)) | bool | 2e94 | 11924 | Not applicable |
| UsrVal.2.Units | Units of the value | string_t | 6902 | 26882 | Not applicable |
| UsrVal.2.Val | Thw User Value | float32 | 2e93 | 11923 | Set by UsrVal.2.Resolution |
| UsrVal.3.HighLimit | User Value High Limit | float32 | 2e96 | 11926 | Set by UsrVal.3.Resolution |
| UsrVal.3.LowLimit | User Value Low Limit | float32 | 2e97 | 11927 | Set by UsrVal.3.Resolution |
| UsrVal.3.Resolution | Result Resolution | uint8 | 2e9a | 11930 | Not applicable |
| UsrVal.3.Status | User Value 3 Status (0 = Good (OK); 7 = Bad (Error)) | bool | 2e99 | 11929 | Not applicable |
| UsrVal.3.Units | Units of the value | string_t | 6908 | 26888 | Not applicable |
| UsrVal.3.Val | The User Value | float32 | 2e98 | 11928 | Set by UsrVal.3.Resolution |
| UsrVal.4.HighLimit | User Value High Limit | float32 | 2e9b | 11931 | Set by UsrVal.4.Resolution |
| UsrVal.4.LowLimit | User Value Low Limit | float32 | 2e9c | 11932 | Set by UsrVal.4.Resolution |
| UsrVal.4.Resolution | Result Resolution | uint8 | 2e9f | 11935 | Not applicable |
| UsrVal.4.Status | User Value 4 Status (0 = Good (OK); 7 = Bad (Error)) | bool | 2e9e | 11934 | Not applicable |
| UsrVal.4.Units | Units of the value | string_t | 690e | 26894 | Not applicable |
| UsrVal.4.Val | The User Value | float32 | 2e9d | 11933 | Set by UsrVal.4.Resolution |
| UsrVal.5.HighLimit | User Value High Limit | float32 | 2ea0 | 11936 | Set by UsrVal.5.Resolution |
| UsrVal.5.LowLimit | User Value Low Limit | float32 | 2ea1 | 11937 | Set by UsrVal.5.Resolution |
| UsrVal.5.Resolution | Result Resolution | uint8 | 2ea4 | 11940 | Not applicable |
| UsrVal.5.Status | User Value 5 Status (0 = Good (OK); 7 = Bad (Error)) | bool | 2ea3 | 11939 | Not applicable |
| UsrVal.5.Units | Units of the value | string_t | 6914 | 26900 | Not applicable |
| UsrVal.5.Val | The User Value | float32 | 2ea2 | 11938 | Set by UsrVal.5.Resolution |
| UsrVal.6.HighLimit | User Value High Limit | float32 | 2ea5 | 11941 | Set by UsrVal.6.Resolution |
| UsrVal.6.LowLimit | User Value Low Limit | float32 | 2ea6 | 11942 | Set by UsrVal.6.Resolution |
| UsrVal.6.Resolution | Result Resolution | uint8 | 2ea9 | 11945 | Not applicable |
| UsrVal.6.Status | User Value 6 Status (0 = Good (OK); 7 = Bad (Error)) | bool | 2ea8 | 11944 | Not applicable |
| UsrVal.6.Units | Units of the value | string_t | 691a | 26906 | Not applicable |
| UsrVal.6.Val | The User Value | float32 | 2ea7 | 11943 | Set by UsrVal.6.Resolution |
| UsrVal.7.HighLimit | User Value High Limit | float32 | 2eaa | 11946 | Set by UsrVal.7.Resolution |
| UsrVal.7.LowLimit | User Value Low Limit | float32 | 2eab | 11947 | Set by UsrVal.7.Resolution |
| UsrVal.7.Resolution | Result Resolution | uint8 | 2eae | 11950 | Not applicable |
| UsrVal.7.Status | User Value 7 Status (0 = Good (OK); 7 = Bad (Error)) | bool | 2ead | 11949 | Not applicable |
| UsrVal.7.Units | Units of the value | string_t | 6920 | 26912 | Not applicable |
| UsrVal.7.Val | The User Value | float32 | 2eac | 11948 | Set by UsrVal.7.Resolution |
| UsrVal.8.HighLimit | User Value High Limit | float32 | 2eaf | 11951 | Set by UsrVal.8.Resolution |
| UsrVal.8.LowLimit | User Value Low Limit | float32 | 2eb0 | 11952 | Set by UsrVal.8.Resolution |
| UsrVal.8.Resolution | Result Resolution | uint8 | 2eb3 | 11955 | Not applicable |
| UsrVal.8.Status | User Value 8 Status (0 = Good (OK); 7 = Bad (Error)) | bool | 2eb2 | 11954 | Not applicable |
| UsrVal.8.Units | Units of the value | string_t | 6926 | 26918 | Not applicable |
| UsrVal.8.Val | The User Value | float32 | 2eb1 | 11953 | Set by UsrVal.8.Resolution |
| UsrVal.9.HighLimit | User Value High Limit | float32 | 2eb4 | 11956 | Set by UsrVal.9.Resolution |
| UsrVal.9.LowLimit | User Value Low Limit | float32 | 2eb5 | 11957 | Set by UsrVal.9.Resolution |
| UsrVal.9.Resolution | Result Resolution | uint8 | 2eb8 | 11960 | Not applicable |
| UsrVal.9.Status | User Value 9 Status (0 = Good (OK); 7 = Bad (Error)) | bool | 2eb7 | 11959 | Not applicable |
| UsrVal.9.Units | Units of the value | string_t | 692c | 26924 | Not applicable |
| UsrVal.9.Val | The User Value | float32 | 2eb6 | 11958 | Set by UsrVal.9.Resolution |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| UsrVal.10.HighLimit | User Value High Limit | float32 | 2eb9 | 11961 | Set by UsrVal.10.Resolution |
| UsrVal.10.LowLimit | User Value Low Limit | float32 | 2eba | 11962 | Set by UsrVal.10.Resolution |
| UsrVal.10.Resolution | Result Resolution | uint8 | 2ebd | 11965 | Not applicable |
| UsrVal.10.Status | User Value 10 Status (0 = Good (OK); 7 = Bad (Error)) | bool | 2ebc | 11964 | Not applicable |
| UsrVal.10.Units | Units of the value | string_t | 6932 | 26930 | Not applicable |
| UsrVal.10.Val | The User Value | float32 | 2ebb | 11963 | Set by UsrVal.10.Resolution |
| UsrVal.11.HighLimit | User Value High Limit | float32 | 2ebe | 11966 | Set by UsrVal.11.Resolution |
| UsrVal.11.LowLimit | User Value Low Limit | float32 | 2ebf | 11967 | Set by UsrVal.11.Resolution |
| UsrVal.11.Resolution | Result Resolution | uint8 | 2ec2 | 11970 | Not applicable |
| UsrVal.11.Status | User Value 11 Status (0 = Good (OK); 7 = Bad (Error)) | bool | 2ec1 | 11969 | Not applicable |
| UsrVal.11.Units | Units of the value | string_t | 6938 | 26936 | Not applicable |
| UsrVal.11.Val | The User Value | float32 | 2ec0 | 11968 | Set by UsrVal.11.Resolution |
| UsrVal.12.HighLimit | User Value High Limit | float32 | 2ec3 | 11971 | Set by UsrVal.12.Resolution |
| UsrVal.12.LowLimit | User Value Low Limit | float32 | 2ec4 | 11972 | Set by UsrVal.12.Resolution |
| UsrVal.12.Resolution | Result Resolution | uint8 | 2ec7 | 11975 | Not applicable |
| UsrVal.12.Status | User Value 12 Status (0 = Good (OK); 7 = Bad (Error)) | bool | 2ec6 | 11974 | Not applicable |
| UsrVal.12.Units | Units of the value | string_t | 693e | 26942 | Not applicable |
| UsrVal.12.Val | The User Value | float32 | 2ec5 | 11973 | Set by UsrVal.12.Resolution |
| VirtualChannel.1.Alarm1.Acknowledge | 1 = acknowledge alarm | bool | 01c0 | 448 | Not applicable |
| VirtualChannel.1.Alarm1.Acknowledgement | 1 = alarm acknowledged | bool | 1c50 | 7248 | Not applicable |
| VirtualChannel.1.Alarm1.Active | 1 = alarm source active, or safe but not ack'd | bool | 1c4b | 7243 | Not applicable |
| VirtualChannel.1.Alarm1.Amount | Rate-of-change alarm 'Amount' | float32 | 1c48 | 7240 | Same as VirtualChannel.1.Main.PV |
| VirtualChannel.1.Alarm1.AverageTime | Rate-of-change alarm 'Average time' | time_t | 1c4a | 7242 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.1.Alarm1.Block | 0 = Blocking alarms off; 1 = Blocking alarms on | uint8 | 1c42 | 7234 | Not applicable |
| VirtualChannel.1.Alarm1.ChangeTime | Rate-of-change alarm 'Change Time' | uint8 | 1c49 | 7241 | Not applicable |
| VirtualChannel.1.Alarm1.Deviation | Deviation alarm 'Deviation Value' | float32 | 1c47 | 7239 | Same as VirtualChannel.1.Main.PV |
| VirtualChannel.1.Alarm1.Dwell | Alarm dwell time | time_t | 1c45 | 7237 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.1.Alarm1.Hysteresis | Alarm hysteresis value | float32 | 1c44 | 7236 | Same as VirtualChannel.1.Main.PV |
| VirtualChannel.1.Alarm1.Inactive | 1 = alarm source safe and ack'd (if necessary) | bool | 1c4e | 7246 | Not applicable |
| VirtualChannel.1.Alarm1.Inhibit | 1 = alarm inhibited | bool | 1c51 | 7249 | Not applicable |
| VirtualChannel.1.Alarm1.Latch | Alarm latch type (0 = None; 1 = Auto; 2 = Manual; 3 = Trigger | uint8 | 1c41 | 7233 | Not applicable |
| VirtualChannel.1.Alarm1.NotAcknowledged | 1 = alarm has not been acknowledged | bool | 1c4f | 7247 | Not applicable |
| VirtualChannel.1.Alarm1.Reference | Deviation alarm 'Reference' value | float32 | 1c46 | 7238 | Same as VirtualChannel.1.Main.PV |
| VirtualChannel.1.Alarm1.Status | Indication of the active and acknowledge status<br>0 = Unacknowledged 1 = None<br>2 = Active 3 = Inactive<br>4 = Acknowledged | uint8 | 0122 | 290 | Not applicable |
| VirtualChannel.1.Alarm1.Threshold | Alarm trigger threshold | float32 | 1c43 | 7235 | Same as VirtualChannel.1.Main.PV |
| VirtualChannel.1.Alarm1.Type | Alarm type<br>0 = None 1 = Abs High 2 = Abs Low<br>3 = Dev high 4 = Dev Low 5 = Dev band<br>6 = ROC rising 7 = ROC falling 10 = Dig Off<br>11 = Dig High 12 = Dig Low | uint8 | 1c40 | 7232 | Not applicable |
| VirtualChannel.1.Alarm2.Acknowledge | 1 = acknowledge alarm | bool | 01c1 | 449 | Not applicable |
| VirtualChannel.1.Alarm2.Acknowledgement | 1 = alarm acknowledged | bool | 1c70 | 7280 | Not applicable |
| VirtualChannel.1.Alarm2.Active | 1 = alarm source active, or safe but not ack'd | bool | 1c6b | 7275 | Not applicable |
| VirtualChannel.1.Alarm2.Amount | Rate-of-change alarm 'Amount' | float32 | 1c68 | 7272 | Same as VirtualChannel.1.Main.PV |
| VirtualChannel.1.Alarm2.AverageTime | Rate-of-change alarm 'Average time' | time_t | 1c6a | 7274 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.1.Alarm2.Block | 0 = Blocking alarms off; 1 = Blocking alarms on | uint8 | 1c62 | 7266 | Not applicable |
| VirtualChannel.1.Alarm2.ChangeTime | Rate-of-change alarm 'Change Time | uint8 | 1c69 | 7273 | Not applicable |
| VirtualChannel.1.Alarm2.Deviation | Deviation alarm 'Deviation Value' | float32 | 1c67 | 7271 | Same as VirtualChannel.1.Main.PV |
| VirtualChannel.1.Alarm2.Dwell | Alarm dwell time | time_t | 1c65 | 7269 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.1.Alarm2.Hysteresis | Alarm hysteresis value | float32 | 1c64 | 7268 | Same as VirtualChannel.1.Main.PV |
| VirtualChannel.1.Alarm2.Inactive | 1 = alarm source safe and ack'd (if necessary) | bool | 1c6e | 7278 | Not applicable |
| VirtualChannel.1.Alarm2.Inhibit | 1 = alarm inhibited | bool | 1c71 | 7281 | Not applicable |
| VirtualChannel.1.Alarm2.Latch | As VirtualChannel1.Alarm1.Latch | uint8 | 1c61 | 7265 | Not applicable |
| VirtualChannel.1.Alarm2.NotAcknowledged | 1 = alarm has not been acknowledged | bool | 1c6f | 7279 | Not applicable |
| VirtualChannel.1.Alarm2.Reference | Deviation alarm 'Reference' value | float32 | 1c66 | 7270 | Same as VirtualChannel.1.Main.PV |
| VirtualChannel.1.Alarm2.Status | As VirtualChannel1.Alarm1.Status | uint8 | 0123 | 291 | Not applicable |
| VirtualChannel.1.Alarm2.Threshold | Alarm trigger threshold | float32 | 1c63 | 7267 | Same as VirtualChannel.1.Main.PV |
| VirtualChannel.1.Alarm2.Type | As VirtualChannel1.Alarm1.Type | uint8 | 1c60 | 7264 | Not applicable |
| VirtualChannel.1.Main.Descriptor | Virtual Channel descriptor | string_t | 4b00 | 19200 | Not applicable |
| VirtualChannel.1.Main.Disable | 1 = Virtual channel disabled | bool | 1c23 | 7203 | Not applicable |
| VirtualChannel.1.Main.HighCutOff | High cut off value for totalisers and counters | float32 | 1c05 | 7173 | Set by VirtualChannel.1.Main.Resolution |
| VirtualChannel.1.Main.Input1 | Input 1 value | float32 | 1c07 | 7175 | Set by VirtualChannel.1.Main.Resolution |
| VirtualChannel.1.Main.Input2 | Input 2 value | float32 | 1c08 | 7176 | Set by VirtualChannel.1.Main.Resolution |
| VirtualChannel.1.Main.LowCutOff | Low cutoff value for totalisers and counters | float32 | 1c04 | 7172 | Set by VirtualChannel.1.Main.Resolution |
| VirtualChannel.1.Main.ModbusInput | Modbus input value | float32 | 1c06 | 7174 | Set by VirtualChannel.1.Main.Resolution |
| VirtualChannel.1.Main.Operation | Specifies the operation of the virtual channel<br>0 = Off 2 = Add 3 = Subtract<br>4 = Multiply 5 = Divide 6 = Group avg<br>7 = Group min 8 = Group max 9 = Modbus i/p<br>11 = Copy 20 = Grp min latch 21 = Grp max latch | uint8 | 1c01 | 7169 | Not applicable |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| | 34 = Chan max 35 = Chan min 36 = Chan avg | | | | |
| | 43 = Config rev 64 = Off 65 = On | | | | |
| | 80 = Off 81 = On | | | | |
| VirtualChannel.1.Main.Period | The time period over which the calculation is made | int32 | 1c0a | 7178 | Not applicable |
| VirtualChannel.1.Main.Preset | Initiate preset. 0 = No; 1 = Yes | bool | 1c0c | 7180 | Not applicable |
| VirtualChannel.1.Main.PresetValue | The preset value | float32 | 1c0d | 7181 | Set by VirtualChannel.1.Main.Resolution |
| VirtualChannel.1.Main.PV | The virtual channel output value | float32 | 0120 | 288 | Set by VirtualChannel.1.Main.Resolution |
| VirtualChannel.1.Main.Reset | Initiate reset. 0 = No; 1 = Yes | bool | 1c0b | 7179 | Not applicable |
| VirtualChannel.1.Main.Resolution | Number of decimal places (0 to 6) | uint8 | 1c02 | 7170 | Not applicable |
| VirtualChannel.1.Main.Rollover | A pulse signal to indicate PV (output) has just rolled over | bool | 1c11 | 7185 | Not applicable |
| VirtualChannel.1.Main.RolloverValue | Rollover value | float32 | 1c12 | 7186 | Set by VirtualChannel.1.Main.Resolution |
| VirtualChannel.1.Main.Status | Virtual Channel output status | uint8 | 0121 | 289 | Not applicable |
| | 0 =Good 1 = Off 2 = Over range | | | | |
| | 3 = Under range 4 = HW error 5 = Ranging | | | | |
| | 6 = Overflow 7 = bad 8 = HW exceeded | | | | |
| | 9 = No data 12 = Comms channel error | | | | |
| VirtualChannel.1.Main.TimeRemaining | Time remaining before the calculation is made | time_t | 1c09 | 7177 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.1.Main.Trigger | Increment/decrement counter. 0 = No; 1 = Yes | bool | 1c0e | 7182 | Not applicable |
| VirtualChannel.1.Main.Type | Specifies the type of virtual channel | uint8 | 1c00 | 7168 | Not applicable |
| | 1 = Maths; 2 = Totaliser; 3 = Counter | | | | |
| VirtualChannel.1.Main.Units | Units descriptor | string_t | 4b15 | 19221 | Not applicable |
| VirtualChannel.1.Main.UnitsScaler | Units scaler for totalisers | float32 | 1c03 | 7171 | 1dp |
| VirtualChannel.1.Trend.Colour | Configures the trend colour for this virtual channel | uint8 | 1c20 | 7200 | Not applicable |
| | 0 = Red 1 = Blue 2 = Green | | | | |
| | 3 = Honey 4 = Violet 5 = Russet | | | | |
| | 6 = Dark blue 7 = Jade 8 = Magenta | | | | |
| | 9 = Dusky rose 10 = Yellow 11 = Powder blue | | | | |
| | 12 = Dark red 13 = Avocado 14 = Indigo | | | | |
| | 15 = Dark brown 16 = Aegean 17 = Cyan | | | | |
| | 18 = Aubergine 19 = Dark orange 20 = Pale yellow | | | | |
| | 21 = Hyacinth 22 = Dark green 23 = Sugar pink | | | | |
| | 24 = Bluebell 25 = Orange 26 = Pink | | | | |
| | 27 = Buttersilk 28 = Terracotta 29 = Blue babe | | | | |
| | 30 = Lime 31 = Blue jive 32 = Cucumber | | | | |
| | 33 = Eurogreen 34 = Wheatgerm 35 = Sea Blue | | | | |
| | 36 = Ginger 37 = Aqua pool 38 = Pale red | | | | |
| | 39 = Pale blue 40 = Lilac 41 = Sky blue | | | | |
| | 42 = Wild moss 43 = Turquoise 44 = Pale green | | | | |
| | 45 = Coffee 49 = Dark Grey 53 = Light grey | | | | |
| VirtualChannel.1.Trend.SpanHigh | Specifies the highest PV (output value) to be displayed | float32 | 1c22 | 7202 | Same as VirtualChannel.1.Main.PV |
| VirtualChannel.1.Trend.SpanLow | Specifies the lowest PV (output value) to be displayed | float32 | 1c21 | 7201 | Same as VirtualChannel.1.Main.PV |
| VirtualChannel.2.Alarm1.Acknowledge | 1 = acknowledge alarm | bool | 01c2 | 450 | Not applicable |
| VirtualChannel.2.Alarm1.Acknowledgement | 1 = alarm acknowledged | bool | 1cd0 | 7376 | Not applicable |
| VirtualChannel.2.Alarm1.Active | 1 = alarm source active, or safe but not ack'd | bool | 1ccb | 7371 | Not applicable |
| VirtualChannel.2.Alarm1.Amount | Rate-of-change alarm 'Amount' | float32 | 1cc8 | 7368 | Same as VirtualChannel.2.Main.PV |
| VirtualChannel.2.Alarm1.AverageTime | Rate-of-change alarm 'Average time' | time_t | 1cca | 7370 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.2.Alarm1.Block | 0 = Blocking alarms off; 1 = Blocking alarms on | uint8 | 1cc2 | 7362 | Not applicable |
| VirtualChannel.2.Alarm1.ChangeTime | Rate-of-change alarm 'Change Time' | uint8 | 1cc9 | 7369 | Not applicable |
| VirtualChannel.2.Alarm1.Deviation | Deviation alarm 'Deviation Value' | float32 | 1cc7 | 7367 | Same as VirtualChannel.2.Main.PV |
| VirtualChannel.2.Alarm1.Dwell | Alarm dwell time | time_t | 1cc5 | 7365 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.2.Alarm1.Hysteresis | Alarm hysteresis value | float32 | 1cc4 | 7364 | Same as VirtualChannel.2.Main.PV |
| VirtualChannel.2.Alarm1.Inactive | 1 = alarm source safe and ack'd (if necessary) | bool | 1cce | 7374 | Not applicable |
| VirtualChannel.2.Alarm1.Inhibit | 1 = alarm inhibited | bool | 1cd1 | 7377 | Not applicable |
| VirtualChannel.2.Alarm1.Latch | As VirtualChannel1.Alarm1.Latch | uint8 | 1cc1 | 7361 | Not applicable |
| VirtualChannel.2.Alarm1.NotAcknowledged | 1 = alarm has not been acknowledged | bool | 1ccf | 7375 | Not applicable |
| VirtualChannel.2.Alarm1.Reference | Deviation alarm 'Reference' value | float32 | 1cc6 | 7366 | Same as VirtualChannel.2.Main.PV |
| VirtualChannel.2.Alarm1.Status | As VirtualChannel1.Alarm1.Status | uint8 | 0126 | 294 | Not applicable |
| VirtualChannel.2.Alarm1.Threshold | Alarm trigger threshold | float32 | 1cc3 | 7363 | Same as VirtualChannel.2.Main.PV |
| VirtualChannel.2.Alarm1.Type | As VirtualChannel1.Alarm1.Type | uint8 | 1cc0 | 7360 | Not applicable |
| VirtualChannel.2.Alarm2.Acknowledge | 1 = acknowledge alarm | bool | 01c3 | 451 | Not applicable |
| VirtualChannel.2.Alarm2.Acknowledgement | 1 = alarm acknowledged | bool | 1cf0 | 7408 | Not applicable |
| VirtualChannel.2.Alarm2.Active | 1 = alarm source active, or safe but not ack'd | bool | 1ceb | 7403 | Not applicable |
| VirtualChannel.2.Alarm2.Amount | Rate-of-change alarm 'Amount' | float32 | 1ce8 | 7400 | Same as VirtualChannel.2.Main.PV |
| VirtualChannel.2.Alarm2.AverageTime | Rate-of-change alarm 'Average time' | time_t | 1cea | 7402 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.2.Alarm2.Block | 0 = Blocking alarms off; 1 = Blocking alarms on | uint8 | 1ce2 | 7394 | Not applicable |
| VirtualChannel.2.Alarm2.ChangeTime | Rate-of-change alarm 'Change Time' | uint8 | 1ce9 | 7401 | Not applicable |
| VirtualChannel.2.Alarm2.Deviation | Deviation alarm 'Deviation Value' | float32 | 1ce7 | 7399 | Same as VirtualChannel.2.Main.PV |
| VirtualChannel.2.Alarm2.Dwell | Alarm dwell time | time_t | 1ce5 | 7397 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.2.Alarm2.Hysteresis | Alarm hysteresis value | float32 | 1ce4 | 7396 | Same as VirtualChannel.2.Main.PV |
| VirtualChannel.2.Alarm2.Inactive | 1 = alarm source safe and ack'd (if necessary) | bool | 1cee | 7406 | Not applicable |
| VirtualChannel.2.Alarm2.Inhibit | 1 = alarm inhibited | bool | 1cf1 | 7409 | Not applicable |
| VirtualChannel.2.Alarm2.Latch | As VirtualChannel1.Alarm1.Latch | uint8 | 1ce1 | 7393 | Not applicable |
| VirtualChannel.2.Alarm2.NotAcknowledged | 1 = alarm has not been acknowledged | bool | 1cef | 7407 | Not applicable |
| VirtualChannel.2.Alarm2.Reference | Deviation alarm 'Reference' value | float32 | 1ce6 | 7398 | Same as VirtualChannel.2.Main.PV |
| VirtualChannel.2.Alarm2.Status | As VirtualChannel1.Alarm1.Status | uint8 | 0127 | 295 | Not applicable |
| VirtualChannel.2.Alarm2.Threshold | Alarm trigger threshold | float32 | 1ce3 | 7395 | Same as VirtualChannel.2.Main.PV |
| VirtualChannel.2.Alarm2.Type | As VirtualChannel1.Alarm1.Type | uint8 | 1ce0 | 7392 | Not applicable |
| VirtualChannel.2.Main.Descriptor | Virtual Channel descriptor | string_t | 4b1b | 19227 | Not applicable |
| VirtualChannel.2.Main.Disable | 1 = Virtual channel disabled | bool | 1ca3 | 7331 | Not applicable |
| VirtualChannel.2.Main.HighCutOff | The highest input value that will be totalised/counted | float32 | 1c85 | 7301 | Set by VirtualChannel.2.Main.Resolution |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| VirtualChannel.2.Main.Input1 | Input 1 value | float32 | 1c87 | 7303 | Set by VirtualChannel.2.Main.Resolution |
| VirtualChannel.2.Main.Input2 | Input 2 value | float32 | 1c88 | 7304 | Set by VirtualChannel.2.Main.Resolution |
| VirtualChannel.2.Main.LowCutOff | The lowest input value that will be totalised/counted | float32 | 1c84 | 7300 | Set by VirtualChannel.2.Main.Resolution |
| VirtualChannel.2.Main.ModbusInput | Modbus input value | float32 | 1c86 | 7302 | Set by VirtualChannel.2.Main.Resolution |
| VirtualChannel.2.Main.Operation | As VirtualChannel1.Main.Operation | uint8 | 1c81 | 7297 | Not applicable |
| VirtualChannel.2.Main.Period | The time period over which the calculation is made | int32 | 1c8a | 7306 | Not applicable |
| VirtualChannel.2.Main.Preset | Initiate preset. 0 = No; 1 = Yes | bool | 1c8c | 7308 | Not applicable |
| VirtualChannel.2.Main.PresetValue | The Preset value | float32 | 1c8d | 7309 | Set by VirtualChannel.2.Main.Resolution |
| VirtualChannel.2.Main.PV | The virtual channel output value | float32 | 0124 | 292 | Set by VirtualChannel.2.Main.Resolution |
| VirtualChannel.2.Main.Reset | Initiate reset. 0 = No; 1 = Yes | bool | 1c8b | 7307 | Not applicable |
| VirtualChannel.2.Main.Resolution | Specifies the resolution/number of decimal places | uint8 | 1c82 | 7298 | Not applicable |
| VirtualChannel.2.Main.Rollover | A pulse signal to indicate PV (output) has just rolled over | bool | 1c91 | 7313 | Not applicable |
| VirtualChannel.2.Main.RolloverValue | Rollover value | float32 | 1c92 | 7314 | Set by VirtualChannel.2.Main.Resolution |
| VirtualChannel.2.Main.Status | As VirtualChannel1.Main.Status | uint8 | 0125 | 293 | Not applicable |
| VirtualChannel.2.Main.TimeRemaining | Time remaining before the calculation is made | time_t | 1c89 | 7305 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.2.Main.Trigger | Increment/decrement counter. 0 = No; 1 = Yes | bool | 1c8e | 7310 | Not applicable |
| VirtualChannel.2.Main.Type | As VirtualChannel1.Main.Type | uint8 | 1c80 | 7296 | Not applicable |
| VirtualChannel.2.Main.Units | Units descriptor | string_t | 4b30 | 19248 | Not applicable |
| VirtualChannel.2.Main.UnitsScaler | Units scaler for totalisers | float32 | 1c83 | 7299 | 1dp |
| VirtualChannel.2.Trend.Colour | As VirtualChannel1.Trend.Colour | uint8 | 1ca0 | 7328 | Not applicable |
| VirtualChannel.2.Trend.SpanHigh | Specifies the highest PV (output value) to be displayed | float32 | 1ca2 | 7330 | Same as VirtualChannel.2.Main.PV |
| VirtualChannel.2.Trend.SpanLow | Specifies the lowest PV (output value) to be displayed | float32 | 1ca1 | 7329 | Same as VirtualChannel.2.Main.PV |
| VirtualChannel.3.Alarm1.Acknowledge | 1 = acknowledge alarm | bool | 01c4 | 452 | Not applicable |
| VirtualChannel.3.Alarm1.Acknowledgement | 1 = alarm acknowledged | bool | 1d50 | 7504 | Not applicable |
| VirtualChannel.3.Alarm1.Active | 1 = alarm source active, or safe but not ack'd | bool | 1d4b | 7499 | Not applicable |
| VirtualChannel.3.Alarm1.Amount | Rate-of-change alarm 'Amount' | float32 | 1d48 | 7496 | Same as VirtualChannel.3.Main.PV |
| VirtualChannel.3.Alarm1.AverageTime | Rate-of-change alarm 'Average time' | time_t | 1d4a | 7498 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.3.Alarm1.Block | 0 = Blocking alarms off; 1 = Blocking alarms on | uint8 | 1d42 | 7490 | Not applicable |
| VirtualChannel.3.Alarm1.ChangeTime | Rate-of-change alarm 'Change Time' | uint8 | 1d49 | 7497 | Not applicable |
| VirtualChannel.3.Alarm1.Deviation | Deviation alarm 'Deviation Value' | float32 | 1d47 | 7495 | Same as VirtualChannel.3.Main.PV |
| VirtualChannel.3.Alarm1.Dwell | Alarm dwell time | time_t | 1d45 | 7493 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.3.Alarm1.Hysteresis | Alarm hysteresis value | float32 | 1d44 | 7492 | Same as VirtualChannel.3.Main.PV |
| VirtualChannel.3.Alarm1.Inactive | 1 = alarm source safe and ack'd (if necessary) | bool | 1d4e | 7502 | Not applicable |
| VirtualChannel.3.Alarm1.Inhibit | 1 = alarm inhibited | bool | 1d51 | 7505 | Not applicable |
| VirtualChannel.3.Alarm1.Latch | As VirtualChannel1.Alarm1.Latch | uint8 | 1d41 | 7489 | Not applicable |
| VirtualChannel.3.Alarm1.NotAcknowledged | 1 = alarm has not been acknowledged | bool | 1d4f | 7503 | Not applicable |
| VirtualChannel.3.Alarm1.Reference | Deviation alarm 'Reference' value | float32 | 1d46 | 7494 | Same as VirtualChannel.3.Main.PV |
| VirtualChannel.3.Alarm1.Status | As VirtualChannel1.Alarm1.Status | uint8 | 012a | 298 | Not applicable |
| VirtualChannel.3.Alarm1.Threshold | Alarm trigger threshold | float32 | 1d43 | 7491 | Same as VirtualChannel.3.Main.PV |
| VirtualChannel.3.Alarm1.Type | As VirtualChannel1.Alarm1.Type | uint8 | 1d40 | 7488 | Not applicable |
| VirtualChannel.3.Alarm2.Acknowledge | 1 = acknowledge alarm | bool | 01c5 | 453 | Not applicable |
| VirtualChannel.3.Alarm2.Acknowledgement | 1 = alarm acknowledged | bool | 1d70 | 7536 | Not applicable |
| VirtualChannel.3.Alarm2.Active | 1 = alarm source active, or safe but not ack'd | bool | 1d6b | 7531 | Not applicable |
| VirtualChannel.3.Alarm2.Amount | Rate-of-change alarm 'Amount' | float32 | 1d68 | 7528 | Same as VirtualChannel.3.Main.PV |
| VirtualChannel.3.Alarm2.AverageTime | Rate-of-change alarm 'Average time' | time_t | 1d6a | 7530 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.3.Alarm2.Block | 0 = Blocking alarms off; 1 = Blocking alarms on | uint8 | 1d62 | 7522 | Not applicable |
| VirtualChannel.3.Alarm2.ChangeTime | Rate-of-change alarm 'Change Time' | uint8 | 1d69 | 7529 | Not applicable |
| VirtualChannel.3.Alarm2.Deviation | Deviation alarm 'Deviation Value' | float32 | 1d67 | 7527 | Same as VirtualChannel.3.Main.PV |
| VirtualChannel.3.Alarm2.Dwell | Alarm dwell time | time_t | 1d65 | 7525 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.3.Alarm2.Hysteresis | Alarm hysteresis value | float32 | 1d64 | 7524 | Same as VirtualChannel.3.Main.PV |
| VirtualChannel.3.Alarm2.Inactive | 1 = alarm source safe and ack'd (if necessary) | bool | 1d6e | 7534 | Not applicable |
| VirtualChannel.3.Alarm2.Inhibit | 1 = alarm inhibited | bool | 1d71 | 7537 | Not applicable |
| VirtualChannel.3.Alarm2.Latch | As VirtualChannel1.Alarm1.Latch | uint8 | 1d61 | 7521 | Not applicable |
| VirtualChannel.3.Alarm2.NotAcknowledged | 1 = alarm has not been acknowledged | bool | 1d6f | 7535 | Not applicable |
| VirtualChannel.3.Alarm2.Reference | Deviation alarm 'Reference' value | float32 | 1d66 | 7526 | Same as VirtualChannel.3.Main.PV |
| VirtualChannel.3.Alarm2.Status | As VirtualChannel1.Alarm1.Status | uint8 | 012b | 299 | Not applicable |
| VirtualChannel.3.Alarm2.Threshold | Alarm trigger threshold | float32 | 1d63 | 7523 | Same as VirtualChannel.3.Main.PV |
| VirtualChannel.3.Alarm2.Type | As VirtualChannel1.Alarm1.Type | uint8 | 1d60 | 7520 | Not applicable |
| VirtualChannel.3.Main.Descriptor | Virtual Channel descriptor | string_t | 4b36 | 19254 | Not applicable |
| VirtualChannel.3.Main.Disable | 1 = Virtual channel disabled | bool | 1d23 | 7459 | Not applicable |
| VirtualChannel.3.Main.HighCutOff | The highest input value that will be totalised/counted | float32 | 1d05 | 7429 | Set by VirtualChannel.3.Main.Resolution |
| VirtualChannel.3.Main.Input1 | Input 1 | float32 | 1d07 | 7431 | Set by VirtualChannel.3.Main.Resolution |
| VirtualChannel.3.Main.Input2 | Input 2 | float32 | 1d08 | 7432 | Set by VirtualChannel.3.Main.Resolution |
| VirtualChannel.3.Main.LowCutOff | The lowest input value that will be totalised/counted | float32 | 1d04 | 7428 | Set by VirtualChannel.3.Main.Resolution |
| VirtualChannel.3.Main.ModbusInput | Modbus input value | float32 | 1d06 | 7430 | Set by VirtualChannel.3.Main.Resolution |
| VirtualChannel.3.Main.Operation | As VirtualChannel1.Main.Operation | uint8 | 1d01 | 7425 | Not applicable |
| VirtualChannel.3.Main.Period | The time period over which the calculation is made | int32 | 1d0a | 7434 | Not applicable |
| VirtualChannel.3.Main.Preset | Initiate preset. 0 = No; 1 = Yes | bool | 1d0c | 7436 | Not applicable |
| VirtualChannel.3.Main.PresetValue | The Preset value | float32 | 1d0d | 7437 | Set by VirtualChannel.3.Main.Resolution |
| VirtualChannel.3.Main.PV | The virtual channel output value | float32 | 0128 | 296 | Set by VirtualChannel.3.Main.Resolution |
| VirtualChannel.3.Main.Reset | Initiate reset. 0 = No; 1 = Yes | bool | 1d0b | 7435 | Not applicable |
| VirtualChannel.3.Main.Resolution | Number of decimal places (0 to 6) | uint8 | 1d02 | 7426 | Not applicable |
| VirtualChannel.3.Main.Rollover | A pulse signal to indicate PV (output) has just rolled over | bool | 1d11 | 7441 | Not applicable |
| VirtualChannel.3.Main.RolloverValue | Rollover value | float32 | 1d12 | 7442 | Set by VirtualChannel.3.Main.Resolution |
| VirtualChannel.3.Main.Status | As VirtualChannel1.Main.Status | uint8 | 0129 | 297 | Not applicable |
| VirtualChannel.3.Main.TimeRemaining | Time remaining before the calculation is made | time_t | 1d09 | 7433 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.3.Main.Trigger | Increment/decrement counter. 0 = No; 1 = Yes | bool | 1d0e | 7438 | Not applicable |
| VirtualChannel.3.Main.Type | As VirtualChannel1.Main.Type | uint8 | 1d00 | 7424 | Not applicable |
| VirtualChannel.3.Main.Units | Units descriptor | string_t | 4b4b | 19275 | Not applicable |
| VirtualChannel.3.Main.UnitsScaler | Units scaler for totalisers | float32 | 1d03 | 7427 | 1dp |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| VirtualChannel.3.Trend.Colour | As VirtualChannel1.Trend.Colour | uint8 | 1d20 | 7456 | Not applicable |
| VirtualChannel.3.Trend.SpanHigh | Specifies the highest PV (output value) to be displayed | float32 | 1d22 | 7458 | Same as VirtualChannel.3.Main.PV |
| VirtualChannel.3.Trend.SpanLow | Specifies the lowest PV (output value) to be displayed | float32 | 1d21 | 7457 | Same as VirtualChannel.3.Main.PV |
| VirtualChannel.4.Alarm1.Acknowledge | 1 = acknowledge alarm | bool | 01c6 | 454 | Not applicable |
| VirtualChannel.4.Alarm1.Acknowledgement | 1 = alarm acknowledged | bool | 1dd0 | 7632 | Not applicable |
| VirtualChannel.4.Alarm1.Active | 1 = alarm source active, or safe but not ack'd | bool | 1dcb | 7627 | Not applicable |
| VirtualChannel.4.Alarm1.Amount | Rate-of-change alarm 'Amount' | float32 | 1dc8 | 7624 | Same as VirtualChannel.4.Main.PV |
| VirtualChannel.4.Alarm1.AverageTime | Rate-of-change alarm 'Average time' | time_t | 1dca | 7626 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.4.Alarm1.Block | 0 = Blocking alarms off; 1 = Blocking alarms on | uint8 | 1dc2 | 7618 | Not applicable |
| VirtualChannel.4.Alarm1.ChangeTime | Rate-of-change alarm 'Change Time' | uint8 | 1dc9 | 7625 | Not applicable |
| VirtualChannel.4.Alarm1.Deviation | Deviation alarm 'Deviation Value' | float32 | 1dc7 | 7623 | Same as VirtualChannel.4.Main.PV |
| VirtualChannel.4.Alarm1.Dwell | Alarm dwell time | time_t | 1dc5 | 7621 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.4.Alarm1.Hysteresis | Alarm hysteresis value | float32 | 1dc4 | 7620 | Same as VirtualChannel.4.Main.PV |
| VirtualChannel.4.Alarm1.Inactive | 1 = alarm source safe and ack'd (if necessary) | bool | 1dce | 7630 | Not applicable |
| VirtualChannel.4.Alarm1.Inhibit | 1 = alarm inhibited | bool | 1dd1 | 7633 | Not applicable |
| VirtualChannel.4.Alarm1.Latch | As VirtualChannel1.Alarm1.Latch | uint8 | 1dc1 | 7617 | Not applicable |
| VirtualChannel.4.Alarm1.NotAcknowledged | 1 = alarm has not been acknowledged | bool | 1dcf | 7631 | Not applicable |
| VirtualChannel.4.Alarm1.Reference | Deviation alarm 'Reference' value | float32 | 1dc6 | 7622 | Same as VirtualChannel.4.Main.PV |
| VirtualChannel.4.Alarm1.Status | As VirtualChannel1.Alarm1.Status | uint8 | 012e | 302 | Not applicable |
| VirtualChannel.4.Alarm1.Threshold | Alarm trigger threshold | float32 | 1dc3 | 7619 | Same as VirtualChannel.4.Main.PV |
| VirtualChannel.4.Alarm1.Type | As VirtualChannel1.Alarm1.Type | uint8 | 1dc0 | 7616 | Not applicable |
| VirtualChannel.4.Alarm2.Acknowledge | 1 = acknowledge alarm | bool | 01c7 | 455 | Not applicable |
| VirtualChannel.4.Alarm2.Acknowledgement | 1 = alarm acknowledged | bool | 1df0 | 7664 | Not applicable |
| VirtualChannel.4.Alarm2.Active | 1 = alarm source active, or safe but not ack'd | bool | 1deb | 7659 | Not applicable |
| VirtualChannel.4.Alarm2.Amount | Rate-of-change alarm 'Amount' | float32 | 1de8 | 7656 | Same as VirtualChannel.4.Main.PV |
| VirtualChannel.4.Alarm2.AverageTime | Rate-of-change alarm 'Average time' | time_t | 1dea | 7658 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.4.Alarm2.Block | 0 = Blocking alarms off; 1 = Blocking alarms on | uint8 | 1de2 | 7650 | Not applicable |
| VirtualChannel.4.Alarm2.ChangeTime | Rate-of-change alarm 'Change Time' | uint8 | 1de9 | 7657 | Not applicable |
| VirtualChannel.4.Alarm2.Deviation | Deviation alarAlarm dwell timeAlarm dwell | time_t | 1de5 | 7653 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.4.Alarm2.Hysteresis | Alarm hysteresis value | float32 | 1de4 | 7652 | Same as VirtualChannel.4.Main.PV |
| VirtualChannel.4.Alarm2.Inactive | 1 = alarm source safe and ack'd (if necessary) | bool | 1dee | 7662 | Not applicable |
| VirtualChannel.4.Alarm2.Inhibit | 1 = alarm inhibited | bool | 1df1 | 7665 | Not applicable |
| VirtualChannel.4.Alarm2.Latch | As VirtualChannel1.Alarm1.Latch | uint8 | 1de1 | 7649 | Not applicable |
| VirtualChannel.4.Alarm2.NotAcknowledged | 1 = alarm has not been acknowledged | bool | 1def | 7663 | Not applicable |
| VirtualChannel.4.Alarm2.Reference | Deviation alarm 'Reference' value | float32 | 1de6 | 7654 | Same as VirtualChannel.4.Main.PV |
| VirtualChannel.4.Alarm2.Status | As VirtualChannel1.Alarm1.Status | uint8 | 012f | 303 | Not applicable |
| VirtualChannel.4.Alarm2.Threshold | Alarm trigger threshold | float32 | 1de3 | 7651 | Same as VirtualChannel.4.Main.PV |
| VirtualChannel.4.Alarm2.Type | As VirtualChannel1.Alarm1.Type | uint8 | 1de0 | 7648 | Not applicable |
| VirtualChannel.4.Main.Descriptor | Virtual Channel descriptor | string_t | 4b51 | 19281 | Not applicable |
| VirtualChannel.4.Main.Disable | 1 = Virtual channel disabled | bool | 1da3 | 7587 | Not applicable |
| VirtualChannel.4.Main.HighCutOff | The highest input value that will be totalised/counted | float32 | 1d85 | 7557 | Set by VirtualChannel.4.Main.Resolution |
| VirtualChannel.4.Main.Input1 | Input 1 value | float32 | 1d87 | 7559 | Set by VirtualChannel.4.Main.Resolution |
| VirtualChannel.4.Main.Input2 | Input 2 value | float32 | 1d88 | 7560 | Set by VirtualChannel.4.Main.Resolution |
| VirtualChannel.4.Main.LowCutOff | The lowest input value that will be totalised/counted | float32 | 1d84 | 7556 | Set by VirtualChannel.4.Main.Resolution |
| VirtualChannel.4.Main.ModbusInput | Modbus input value | float32 | 1d86 | 7558 | Set by VirtualChannel.4.Main.Resolution |
| VirtualChannel.4.Main.Operation | As VirtualChannel1.Main.Operation | uint8 | 1d81 | 7553 | Not applicable |
| VirtualChannel.4.Main.Period | Averaging period | int32 | 1d8a | 7562 | Not applicable |
| VirtualChannel.4.Main.Preset | Initiate preset. 0 = No; 1 = Yes | bool | 1d8c | 7564 | Not applicable |
| VirtualChannel.4.Main.PresetValue | The Preset value | float32 | 1d8d | 7565 | Set by VirtualChannel.4.Main.Resolution |
| VirtualChannel.4.Main.PV | The virtual channel output value | float32 | 012c | 300 | Set by VirtualChannel.4.Main.Resolution |
| VirtualChannel.4.Main.Reset | Initiate reset. 0 = No; 1 = Yes | bool | 1d8b | 7563 | Not applicable |
| VirtualChannel.4.Main.Resolution | Number of decimal places (0 to 6) | uint8 | 1d82 | 7554 | Not applicable |
| VirtualChannel.4.Main.Rollover | A pulse signal to indicate PV (output) has just rolled over | bool | 1d91 | 7569 | Not applicable |
| VirtualChannel.4.Main.RolloverValue | Rollover value | float32 | 1d92 | 7570 | Set by VirtualChannel.4.Main.Resolution |
| VirtualChannel.4.Main.Status | As VirtualChannel1.Main.Status | uint8 | 012d | 301 | Not applicable |
| VirtualChannel.4.Main.TimeRemaining | Time remaining before the calculation is made | time_t | 1d89 | 7561 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.4.Main.Trigger | Increment/decrement counter. 0 = No; 1 = Yes | bool | 1d8e | 7566 | Not applicable |
| VirtualChannel.4.Main.Type | As VirtualChannel1.Main.Type | uint8 | 1d80 | 7552 | Not applicable |
| VirtualChannel.4.Main.Units | Units descriptor | string_t | 4b66 | 19302 | Not applicable |
| VirtualChannel.4.Main.UnitsScaler | Units scaler for totalisers | float32 | 1d83 | 7555 | 1dp |
| VirtualChannel.4.Trend.Colour | As VirtualChannel1.Trend.Colour | uint8 | 1da0 | 7584 | Not applicable |
| VirtualChannel.4.Trend.SpanHigh | Specifies the highest PV (output value) to be displayed | float32 | 1da2 | 7586 | Same as VirtualChannel.4.Main.PV |
| VirtualChannel.4.Trend.SpanLow | Specifies the lowest PV (output value) to be displayed | float32 | 1da1 | 7585 | Same as VirtualChannel.4.Main.PV |
| VirtualChannel.5.Alarm1.Acknowledge | 1 = acknowledge alarm | bool | 01c8 | 456 | Not applicable |
| VirtualChannel.5.Alarm1.Acknowledgement | 1 = alarm acknowledged | bool | 1e50 | 7760 | Not applicable |
| VirtualChannel.5.Alarm1.Active | 1 = alarm source active, or safe but not ack'd | bool | 1e4b | 7755 | Not applicable |
| VirtualChannel.5.Alarm1.Amount | Rate-of-change alarm 'Amount' | float32 | 1e48 | 7752 | Same as VirtualChannel.5.Main.PV |
| VirtualChannel.5.Alarm1.AverageTime | Rate-of-change alarm 'Average time' | time_t | 1e4a | 7754 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.5.Alarm1.Block | 0 = Blocking alarms off; 1 = Blocking alarms on | uint8 | 1e42 | 7746 | Not applicable |
| VirtualChannel.5.Alarm1.ChangeTime | Rate-of-change alarm 'Change Time' | uint8 | 1e49 | 7753 | Not applicable |
| VirtualChannel.5.Alarm1.Deviation | Deviation alarm 'Deviation Value' | float32 | 1e47 | 7751 | Same as VirtualChannel.5.Main.PV |
| VirtualChannel.5.Alarm1.Dwell | Alarm dwell time | time_t | 1e45 | 7749 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.5.Alarm1.Hysteresis | Alarm hysteresis value | float32 | 1e44 | 7748 | Same as VirtualChannel.5.Main.PV |
| VirtualChannel.5.Alarm1.Inactive | 1 = alarm source safe and ack'd (if necessary) | bool | 1e4e | 7758 | Not applicable |
| VirtualChannel.5.Alarm1.Inhibit | 1 = alarm inhibited | bool | 1e51 | 7761 | Not applicable |
| VirtualChannel.5.Alarm1.Latch | As VirtualChannel1.Alarm1.Latch | uint8 | 1e41 | 7745 | Not applicable |
| VirtualChannel.5.Alarm1.NotAcknowledged | 1 = alarm has not been acknowledged | bool | 1e4f | 7759 | Not applicable |
| VirtualChannel.5.Alarm1.Reference | Deviation alarm 'Reference' value | float32 | 1e46 | 7750 | Same as VirtualChannel.5.Main.PV |
| VirtualChannel.5.Alarm1.Status | As VirtualChannel1.Alarm1.Status | uint8 | 0132 | 306 | Not applicable |
| VirtualChannel.5.Alarm1.Threshold | Alarm trigger threshold | float32 | 1e43 | 7747 | Same as VirtualChannel.5.Main.PV |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| VirtualChannel.5.Alarm1.Type | As VirtualChannel1.Alarm1.Type | uint8 | 1e40 | 7744 | Not applicable |
| VirtualChannel.5.Alarm2.Acknowledge | 1 = acknowledge alarm | bool | 01c9 | 457 | Not applicable |
| VirtualChannel.5.Alarm2.Acknowledgement | 1 = alarm acknowledged | bool | 1e70 | 7792 | Not applicable |
| VirtualChannel.5.Alarm2.Active | 1 = alarm source active, or safe but not ack'd | bool | 1e6b | 7787 | Not applicable |
| VirtualChannel.5.Alarm2.Amount | Rate-of-change alarm 'Amount' | float32 | 1e68 | 7784 | Same as VirtualChannel.5.Main.PV |
| VirtualChannel.5.Alarm2.AverageTime | Rate-of-change alarm 'Average time' | time_t | 1e6a | 7786 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.5.Alarm2.Block | 0 = Blocking alarms off; 1 = Blocking alarms on | uint8 | 1e62 | 7778 | Not applicable |
| VirtualChannel.5.Alarm2.ChangeTime | Rate-of-change alarm 'Change Time' | uint8 | 1e69 | 7785 | Not applicable |
| VirtualChannel.5.Alarm2.Deviation | Deviation alarm 'Deviation Value' | float32 | 1e67 | 7783 | Same as VirtualChannel.5.Main.PV |
| VirtualChannel.5.Alarm2.Dwell | Alarm dwell time | time_t | 1e65 | 7781 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.5.Alarm2.Hysteresis | Alarm hysteresis value | float32 | 1e64 | 7780 | Same as VirtualChannel.5.Main.PV |
| VirtualChannel.5.Alarm2.Inactive | 1 = alarm source safe and ack'd (if necessary) | bool | 1e6e | 7790 | Not applicable |
| VirtualChannel.5.Alarm2.Inhibit | 1 = alarm inhibited | bool | 1e71 | 7793 | Not applicable |
| VirtualChannel.5.Alarm2.Latch | As VirtualChannel1.Alarm1.Latch | uint8 | 1e61 | 7777 | Not applicable |
| VirtualChannel.5.Alarm2.NotAcknowledged | 1 = alarm has not been acknowledged | bool | 1e6f | 7791 | Not applicable |
| VirtualChannel.5.Alarm2.Reference | Deviation alarm 'Reference' value | float32 | 1e66 | 7782 | Same as VirtualChannel.5.Main.PV |
| VirtualChannel.5.Alarm2.Status | As VirtualChannel1.Alarm1.Status | uint8 | 0133 | 307 | Not applicable |
| VirtualChannel.5.Alarm2.Threshold | Alarm trigger threshold | float32 | 1e63 | 7779 | Same as VirtualChannel.5.Main.PV |
| VirtualChannel.5.Alarm2.Type | As VirtualChannel1.Alarm1.Type | uint8 | 1e60 | 7776 | Not applicable |
| VirtualChannel.5.Main.Descriptor | Virtual Channel descriptor | string_t | 4b6c | 19308 | Not applicable |
| VirtualChannel.5.Main.Disable | 1 = Virtual channel disabled | bool | 1e23 | 7715 | Not applicable |
| VirtualChannel.5.Main.HighCutOff | The highest input value that will be totalised/counted | float32 | 1e05 | 7685 | Set by VirtualChannel.5.Main.Resolution |
| VirtualChannel.5.Main.Input1 | Input 1 value | float32 | 1e07 | 7687 | Set by VirtualChannel.5.Main.Resolution |
| VirtualChannel.5.Main.Input2 | Input 2 value | float32 | 1e08 | 7688 | Set by VirtualChannel.5.Main.Resolution |
| VirtualChannel.5.Main.LowCutOff | The lowest input value that will be totalised/counted | float32 | 1e04 | 7684 | Set by VirtualChannel.5.Main.Resolution |
| VirtualChannel.5.Main.ModbusInput | Modbus input value | float32 | 1e06 | 7686 | Set by VirtualChannel.5.Main.Resolution |
| VirtualChannel.5.Main.Operation | As VirtualChannel1.Main.Operation | uint8 | 1e01 | 7681 | Not applicable |
| VirtualChannel.5.Main.Period | The time period over which the calculation is made | int32 | 1e0a | 7690 | Not applicable |
| VirtualChannel.5.Main.Preset | Initiate preset. 0 = No; 1 = Yes | bool | 1e0c | 7692 | Not applicable |
| VirtualChannel.5.Main.PresetValue | The Preset value | float32 | 1e0d | 7693 | Set by VirtualChannel.5.Main.Resolution |
| VirtualChannel.5.Main.PV | The virtual channel output value | float32 | 0130 | 304 | Set by VirtualChannel.5.Main.Resolution |
| VirtualChannel.5.Main.Reset | nitiate reset. 0 = No; 1 = Yes | bool | 1e0b | 7691 | Not applicable |
| VirtualChannel.5.Main.Resolution | Number of decimal places (0 to 6) | uint8 | 1e02 | 7682 | Not applicable |
| VirtualChannel.5.Main.Rollover | A pulse signal to indicate PV (output) has just rolled over | bool | 1e11 | 7697 | Not applicable |
| VirtualChannel.5.Main.RolloverValue | Rollover value | float32 | 1e12 | 7698 | Set by VirtualChannel.5.Main.Resolution |
| VirtualChannel.5.Main.Status | As VirtualChannel1.Main.Status | uint8 | 0131 | 305 | Not applicable |
| VirtualChannel.5.Main.TimeRemaining | Time remaining before the calculation is made | time_t | 1e09 | 7689 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.5.Main.Trigger | Increment/decrement counter. 0 = No; 1 = Yes | bool | 1e0e | 7694 | Not applicable |
| VirtualChannel.5.Main.Type | As VirtualChannel1.Main.Type | uint8 | 1e00 | 7680 | Not applicable |
| VirtualChannel.5.Main.Units | Units descriptor | string_t | 4b81 | 19329 | Not applicable |
| VirtualChannel.5.Main.UnitsScaler | Units scaler for totalisers | float32 | 1e03 | 7683 | 1dp |
| VirtualChannel.5.Trend.Colour | As VirtualChannel1.Trend.Colour | uint8 | 1e20 | 7712 | Not applicable |
| VirtualChannel.5.Trend.SpanHigh | Specifies the highest PV (output value) to be displayed | float32 | 1e22 | 7714 | Same as VirtualChannel.5.Main.PV |
| VirtualChannel.5.Trend.SpanLow | Specifies the lowest PV (output value) to be displayed | float32 | 1e21 | 7713 | Same as VirtualChannel.5.Main.PV |
| VirtualChannel.6.Alarm1.Acknowledge | 1 = acknowledge alarm | bool | 01ca | 458 | Not applicable |
| VirtualChannel.6.Alarm1.Acknowledgement | 1 = alarm acknowledged | bool | 1ed0 | 7888 | Not applicable |
| VirtualChannel.6.Alarm1.Active | 1 = alarm source active, or safe but not ack'd | bool | 1ecb | 7883 | Not applicable |
| VirtualChannel.6.Alarm1.Amount | Rate-of-change alarm 'Amount' | float32 | 1ec8 | 7880 | Same as VirtualChannel.6.Main.PV |
| VirtualChannel.6.Alarm1.AverageTime | Rate-of-change alarm 'Average time' | time_t | 1eca | 7882 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.6.Alarm1.Block | 0 = Blocking alarms off; 1 = Blocking alarms on | uint8 | 1ec2 | 7874 | Not applicable |
| VirtualChannel.6.Alarm1.ChangeTime | Rate-of-change alarm 'Change Time' | uint8 | 1ec9 | 7881 | Not applicable |
| VirtualChannel.6.Alarm1.Deviation | Deviation alarm 'Deviation Value' | float32 | 1ec7 | 7879 | Same as VirtualChannel.6.Main.PV |
| VirtualChannel.6.Alarm1.Dwell | Alarm dwell time | time_t | 1ec5 | 7877 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.6.Alarm1.Hysteresis | Alarm hysteresis value | float32 | 1ec4 | 7876 | Same as VirtualChannel.6.Main.PV |
| VirtualChannel.6.Alarm1.Inactive | 1 = alarm source safe and ack'd (if necessary) | bool | 1ece | 7886 | Not applicable |
| VirtualChannel.6.Alarm1.Inhibit | 1 = alarm inhibited | bool | 1ed1 | 7889 | Not applicable |
| VirtualChannel.6.Alarm1.Latch | As VirtualChannel1.Alarm1.Latch | uint8 | 1ec1 | 7873 | Not applicable |
| VirtualChannel.6.Alarm1.NotAcknowledged | 1 = alarm has not been acknowledged | bool | 1ecf | 7887 | Not applicable |
| VirtualChannel.6.Alarm1.Reference | Deviation alarm 'Reference' value | float32 | 1ec6 | 7878 | Same as VirtualChannel.6.Main.PV |
| VirtualChannel.6.Alarm1.Status | As VirtualChannel1.Alarm1.Status | uint8 | 0136 | 310 | Not applicable |
| VirtualChannel.6.Alarm1.Threshold | Alarm trigger threshold | float32 | 1ec3 | 7875 | Same as VirtualChannel.6.Main.PV |
| VirtualChannel.6.Alarm1.Type | As VirtualChannel1.Alarm1.Type | uint8 | 1ec0 | 7872 | Not applicable |
| VirtualChannel.6.Alarm2.Acknowledge | 1 = acknowledge alarm | bool | 01cb | 459 | Not applicable |
| VirtualChannel.6.Alarm2.Acknowledgement | 1 = alarm acknowledged | bool | 1ef0 | 7920 | Not applicable |
| VirtualChannel.6.Alarm2.Active | 1 = alarm source active, or safe but not ack'd | bool | 1eeb | 7915 | Not applicable |
| VirtualChannel.6.Alarm2.Amount | Rate-of-change alarm 'Amount' | float32 | 1ee8 | 7912 | Same as VirtualChannel.6.Main.PV |
| VirtualChannel.6.Alarm2.AverageTime | Rate-of-change alarm 'Average time' | time_t | 1eea | 7914 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.6.Alarm2.Block | 0 = Blocking alarms off; 1 = Blocking alarms on | uint8 | 1ee2 | 7906 | Not applicable |
| VirtualChannel.6.Alarm2.ChangeTime | Rate-of-change alarm 'Change Time' | uint8 | 1ee9 | 7913 | Not applicable |
| VirtualChannel.6.Alarm2.Deviation | Deviation alarm 'Deviation Value' | float32 | 1ee7 | 7911 | Same as VirtualChannel.6.Main.PV |
| VirtualChannel.6.Alarm2.Dwell | Alarm dwell time | time_t | 1ee5 | 7909 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.6.Alarm2.Hysteresis | Alarm hysteresis value | float32 | 1ee4 | 7908 | Same as VirtualChannel.6.Main.PV |
| VirtualChannel.6.Alarm2.Inactive | 1 = alarm source safe and ack'd (if necessary) | bool | 1eee | 7918 | Not applicable |
| VirtualChannel.6.Alarm2.Inhibit | 1 = alarm inhibited | bool | 1ef1 | 7921 | Not applicable |
| VirtualChannel.6.Alarm2.Latch | As VirtualChannel1.Alarm1.Latch | uint8 | 1ee1 | 7905 | Not applicable |
| VirtualChannel.6.Alarm2.NotAcknowledged | 1 = alarm has not been acknowledged | bool | 1eef | 7919 | Not applicable |
| VirtualChannel.6.Alarm2.Reference | Deviation alarm 'Reference' value | float32 | 1ee6 | 7910 | Same as VirtualChannel.6.Main.PV |
| VirtualChannel.6.Alarm2.Status | As VirtualChannel1.Alarm1.Status | uint8 | 0137 | 311 | Not applicable |
| VirtualChannel.6.Alarm2.Threshold | Alarm trigger threshold | float32 | 1ee3 | 7907 | Same as VirtualChannel.6.Main.PV |
| VirtualChannel.6.Alarm2.Type | As VirtualChannel1.Alarm1.Type | uint8 | 1ee0 | 7904 | Not applicable |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| VirtualChannel.6.Main.Descriptor | Virtual Channel descriptor | string_t | 4b87 | 19335 | Not applicable |
| VirtualChannel.6.Main.Disable | 1 = Virtual channel disabled | bool | 1ea3 | 7843 | Not applicable |
| VirtualChannel.6.Main.HighCutOff | The highest input value that will be totalised/counted | float32 | 1e85 | 7813 | Set by VirtualChannel.6.Main.Resolution |
| VirtualChannel.6.Main.Input1 | Input 1 value | float32 | 1e87 | 7815 | Set by VirtualChannel.6.Main.Resolution |
| VirtualChannel.6.Main.Input2 | Input 2 value | float32 | 1e88 | 7816 | Set by VirtualChannel.6.Main.Resolution |
| VirtualChannel.6.Main.LowCutOff | The lowest input value that will be totalised/counted | float32 | 1e84 | 7812 | Set by VirtualChannel.6.Main.Resolution |
| VirtualChannel.6.Main.ModbusInput | Modbus input value | float32 | 1e86 | 7814 | Set by VirtualChannel.6.Main.Resolution |
| VirtualChannel.6.Main.Operation | As VirtualChannel1.Main.Operation | uint8 | 1e81 | 7809 | Not applicable |
| VirtualChannel.6.Main.Period | The time period over which the calculation is made | int32 | 1e8a | 7818 | Not applicable |
| VirtualChannel.6.Main.Preset | Initiate preset. 0 = No; 1 = Yes | bool | 1e8c | 7820 | Not applicable |
| VirtualChannel.6.Main.PresetValue | The Preset value | float32 | 1e8d | 7821 | Set by VirtualChannel.6.Main.Resolution |
| VirtualChannel.6.Main.PV | The virtual channel output value | float32 | 0134 | 308 | Set by VirtualChannel.6.Main.Resolution |
| VirtualChannel.6.Main.Reset | Initiate reset. 0 = No; 1 = Yes | bool | 1e8b | 7819 | Not applicable |
| VirtualChannel.6.Main.Resolution | Number of decimal places (0 to 6) | uint8 | 1e82 | 7810 | Not applicable |
| VirtualChannel.6.Main.Rollover | A pulse signal to indicate PV (output) has just rolled over | bool | 1e91 | 7825 | Not applicable |
| VirtualChannel.6.Main.RolloverValue | Rollover value | float32 | 1e92 | 7826 | Set by VirtualChannel.6.Main.Resolution |
| VirtualChannel.6.Main.Status | As VirtualChannel1.Main.Status | uint8 | 0135 | 309 | Not applicable |
| VirtualChannel.6.Main.TimeRemaining | Time remaining before the calculation is made | time_t | 1e89 | 7817 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.6.Main.Trigger | ncrement/decrement counter. 0 = No; 1 = Yes | bool | 1e8e | 7822 | Not applicable |
| VirtualChannel.6.Main.Type | As VirtualChannel1.Main.Type | uint8 | 1e80 | 7808 | Not applicable |
| VirtualChannel.6.Main.Units | Units descriptor | string_t | 4b9c | 19356 | Not applicable |
| VirtualChannel.6.Main.UnitsScaler | Units scaler for totalisers | float32 | 1e83 | 7811 | 1dp |
| VirtualChannel.6.Trend.Colour | As VirtualChannel1.Trend.Colour | uint8 | 1ea0 | 7840 | Not applicable |
| VirtualChannel.6.Trend.SpanHigh | Specifies the highest PV (output value) to be displayed | float32 | 1ea2 | 7842 | Same as VirtualChannel.6.Main.PV |
| VirtualChannel.6.Trend.SpanLow | Specifies the lowest PV (output value) to be displayed | float32 | 1ea1 | 7841 | Same as VirtualChannel.6.Main.PV |
| VirtualChannel.7.Alarm1.Acknowledge | 1 = acknowledge alarm | bool | 01cc | 460 | Not applicable |
| VirtualChannel.7.Alarm1.Acknowledgement | 1 = alarm acknowledged | bool | 1f50 | 8016 | Not applicable |
| VirtualChannel.7.Alarm1.Active | 1 = alarm source active, or safe but not ack'd | bool | 1f4b | 8011 | Not applicable |
| VirtualChannel.7.Alarm1.Amount | Rate-of-change alarm 'Amount' | float32 | 1f48 | 8008 | Same as VirtualChannel.7.Main.PV |
| VirtualChannel.7.Alarm1.AverageTime | Rate-of-change alarm 'Average time' | time_t | 1f4a | 8010 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.7.Alarm1.Block | 0 = Blocking alarms off; 1 = Blocking alarms on | uint8 | 1f42 | 8002 | Not applicable |
| VirtualChannel.7.Alarm1.ChangeTime | Rate-of-change alarm 'Change Time' | uint8 | 1f49 | 8009 | Not applicable |
| VirtualChannel.7.Alarm1.Deviation | Deviation alarm 'Deviation Value' | float32 | 1f47 | 8007 | Same as VirtualChannel.7.Main.PV |
| VirtualChannel.7.Alarm1.Dwell | Alarm dwell time | time_t | 1f45 | 8005 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.7.Alarm1.Hysteresis | Alarm hysteresis value | float32 | 1f44 | 8004 | Same as VirtualChannel.7.Main.PV |
| VirtualChannel.7.Alarm1.Inactive | 1 = alarm source safe and ack'd (if necessary) | bool | 1f4e | 8014 | Not applicable |
| VirtualChannel.7.Alarm1.Inhibit | 1 = alarm inhibited | bool | 1f51 | 8017 | Not applicable |
| VirtualChannel.7.Alarm1.Latch | As VirtualChannel1.Alarm1.Latch | uint8 | 1f41 | 8001 | Not applicable |
| VirtualChannel.7.Alarm1.NotAcknowledged | 1 = alarm has not been acknowledged | bool | 1f4f | 8015 | Not applicable |
| VirtualChannel.7.Alarm1.Reference | Deviation alarm 'Reference' value | float32 | 1f46 | 8006 | Same as VirtualChannel.7.Main.PV |
| VirtualChannel.7.Alarm1.Status | As VirtualChannel1.Alarm1.Status | uint8 | 013a | 314 | Not applicable |
| VirtualChannel.7.Alarm1.Threshold | Alarm trigger threshold | float32 | 1f43 | 8003 | Same as VirtualChannel.7.Main.PV |
| VirtualChannel.7.Alarm1.Type | As VirtualChannel1.Alarm1.Type | uint8 | 1f40 | 8000 | Not applicable |
| VirtualChannel.7.Alarm2.Acknowledge | 1 = acknowledge alarm | bool | 01cd | 461 | Not applicable |
| VirtualChannel.7.Alarm2.Acknowledgement | 1 = alarm acknowledged | bool | 1f70 | 8048 | Not applicable |
| VirtualChannel.7.Alarm2.Active | 1 = alarm source active, or safe but not ack'd | bool | 1f6b | 8043 | Not applicable |
| VirtualChannel.7.Alarm2.Amount | Rate-of-change alarm 'Amount' | float32 | 1f68 | 8040 | Same as VirtualChannel.7.Main.PV |
| VirtualChannel.7.Alarm2.AverageTime | Rate-of-change alarm 'Average time' | time_t | 1f6a | 8042 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.7.Alarm2.Block | 0 = Blocking alarms off; 1 = Blocking alarms on | uint8 | 1f62 | 8034 | Not applicable |
| VirtualChannel.7.Alarm2.ChangeTime | Rate-of-change alarm 'Change Time' | uint8 | 1f69 | 8041 | Not applicable |
| VirtualChannel.7.Alarm2.Deviation | Deviation alarm 'Deviation Value' | float32 | 1f67 | 8039 | Same as VirtualChannel.7.Main.PV |
| VirtualChannel.7.Alarm2.Dwell | Alarm dwell time | time_t | 1f65 | 8037 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.7.Alarm2.Hysteresis | Alarm hysteresis value | float32 | 1f64 | 8036 | Same as VirtualChannel.7.Main.PV |
| VirtualChannel.7.Alarm2.Inactive | 1 = alarm source safe and ack'd (if necessary) | bool | 1f6e | 8046 | Not applicable |
| VirtualChannel.7.Alarm2.Inhibit | 1 = alarm inhibited | bool | 1f71 | 8049 | Not applicable |
| VirtualChannel.7.Alarm2.Latch | As VirtualChannel1.Alarm1.Latch | uint8 | 1f61 | 8033 | Not applicable |
| VirtualChannel.7.Alarm2.NotAcknowledged | 1 = alarm has not been acknowledged | bool | 1f6f | 8047 | Not applicable |
| VirtualChannel.7.Alarm2.Reference | Deviation alarm 'Reference' value | float32 | 1f66 | 8038 | Same as VirtualChannel.7.Main.PV |
| VirtualChannel.7.Alarm2.Status | As VirtualChanneAlarm trigger thresholdAlarm threshold | float32 | 1f63 | 8035 | Same as VirtualChannel.7.Main.PV |
| VirtualChannel.7.Alarm2.Type | As VirtualChannel1.Alarm1.Type | uint8 | 1f60 | 8032 | Not applicable |
| VirtualChannel.7.Main.Descriptor | Virtual Channel descriptor | string_t | 4ba2 | 19362 | Not applicable |
| VirtualChannel.7.Main.Disable | 1 = Virtual channel disabled | bool | 1f23 | 7971 | Not applicable |
| VirtualChannel.7.Main.HighCutOff | The highest input value that will be totalised/counted | float32 | 1f05 | 7941 | Set by VirtualChannel.7.Main.Resolution |
| VirtualChannel.7.Main.Input1 | Input 1 value | float32 | 1f07 | 7943 | Set by VirtualChannel.7.Main.Resolution |
| VirtualChannel.7.Main.Input2 | Input 2 value | float32 | 1f08 | 7944 | Set by VirtualChannel.7.Main.Resolution |
| VirtualChannel.7.Main.LowCutOff | The lowest input value that will be totalised/counted | float32 | 1f04 | 7940 | Set by VirtualChannel.7.Main.Resolution |
| VirtualChannel.7.Main.ModbusInput | Modbus input value | float32 | 1f06 | 7942 | Set by VirtualChannel.7.Main.Resolution |
| VirtualChannel.7.Main.Operation | As VirtualChannel1.Main.Operation | uint8 | 1f01 | 7937 | Not applicable |
| VirtualChannel.7.Main.Period | Averaging period | int32 | 1f0a | 7946 | Not applicable |
| VirtualChannel.7.Main.Preset | Initiate preset. 0 = No; 1 = Yes | bool | 1f0c | 7948 | Not applicable |
| VirtualChannel.7.Main.PresetValue | The Preset value | float32 | 1f0d | 7949 | Set by VirtualChannel.7.Main.Resolution |
| VirtualChannel.7.Main.PV | The virtual channel output value | float32 | 0138 | 312 | Set by VirtualChannel.7.Main.Resolution |
| VirtualChannel.7.Main.Reset | Initiate reset. 0 = No; 1 = Yes | bool | 1f0b | 7947 | Not applicable |
| VirtualChannel.7.Main.Resolution | Number of decimal places (0 to 6) | uint8 | 1f02 | 7938 | Not applicable |
| VirtualChannel.7.Main.Rollover | A pulse signal to indicate PV (output) has just rolled over | bool | 1f11 | 7953 | Not applicable |
| VirtualChannel.7.Main.RolloverValue | Rollover value | float32 | 1f12 | 7954 | Set by VirtualChannel.7.Main.Resolution |
| VirtualChannel.7.Main.Status | As VirtualChannel1.Main.Status | uint8 | 0139 | 313 | Not applicable |
| VirtualChannel.7.Main.TimeRemaining | Time remaining before calculation is made | time_t | 1f09 | 7945 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.7.Main.Trigger | Increment/decrement counter. 0 = No; 1 = Yes | bool | 1f0e | 7950 | Not applicable |
| VirtualChannel.7.Main.Type | As VirtualChannel1.Main.Type | uint8 | 1f00 | 7936 | Not applicable |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| VirtualChannel.7.Main.Units | Units descriptor | string_t | 4bb7 | 19383 | Not applicable |
| VirtualChannel.7.Main.UnitsScaler | Units scaler for totalisers | float32 | 1f03 | 7939 | 1dp |
| VirtualChannel.7.Trend.Colour | As VirtualChannel1.Trend.Colour | uint8 | 1f20 | 7968 | Not applicable |
| VirtualChannel.7.Trend.SpanHigh | Specifies the highest PV (output value) to be displayed | float32 | 1f22 | 7970 | Same as VirtualChannel.7.Main.PV |
| VirtualChannel.7.Trend.SpanLow | Specifies the lowest PV (output value) to be displayed | float32 | 1f21 | 7969 | Same as VirtualChannel.7.Main.PV |
| VirtualChannel.8.Alarm1.Acknowledge | 1 = acknowledge alarm | bool | 01ce | 462 | Not applicable |
| VirtualChannel.8.Alarm1.Acknowledgement | 1 = alarm acknowledged | bool | 1fd0 | 8144 | Not applicable |
| VirtualChannel.8.Alarm1.Active | 1 = alarm source active, or safe but not ack'd | bool | 1fcb | 8139 | Not applicable |
| VirtualChannel.8.Alarm1.Amount | Rate-of-change alarm 'Amount' | float32 | 1fc8 | 8136 | Same as VirtualChannel.8.Main.PV |
| VirtualChannel.8.Alarm1.AverageTime | Rate-of-change alarm 'Average time' | time_t | 1fca | 8138 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.8.Alarm1.Block | 0 = Blocking alarms off; 1 = Blocking alarms on | uint8 | 1fc2 | 8130 | Not applicable |
| VirtualChannel.8.Alarm1.ChangeTime | Rate-of-change alarm 'Change Time' | uint8 | 1fc9 | 8137 | Not applicable |
| VirtualChannel.8.Alarm1.Deviation | Deviation alarm 'Deviation Value' | float32 | 1fc7 | 8135 | Same as VirtualChannel.8.Main.PV |
| VirtualChannel.8.Alarm1.Dwell | Alarm dwell time | time_t | 1fc5 | 8133 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.8.Alarm1.Hysteresis | Alarm hysteresis value | float32 | 1fc4 | 8132 | Same as VirtualChannel.8.Main.PV |
| VirtualChannel.8.Alarm1.Inactive | 1 = alarm source safe and ack'd (if necessary) | bool | 1fce | 8142 | Not applicable |
| VirtualChannel.8.Alarm1.Inhibit | 1 = alarm inhibited | bool | 1fd1 | 8145 | Not applicable |
| VirtualChannel.8.Alarm1.Latch | As VirtualChannel1.Alarm1.Latch | uint8 | 1fc1 | 8129 | Not applicable |
| VirtualChannel.8.Alarm1.NotAcknowledged | 1 = alarm has not been acknowledged | bool | 1fcf | 8143 | Not applicable |
| VirtualChannel.8.Alarm1.Reference | Deviation alarm 'Reference' value | float32 | 1fc6 | 8134 | Same as VirtualChannel.8.Main.PV |
| VirtualChannel.8.Alarm1.Status | As VirtualChannel1.Alarm1.Status | uint8 | 013e | 318 | Not applicable |
| VirtualChannel.8.Alarm1.Threshold | Alarm trigger threshold | float32 | 1fc3 | 8131 | Same as VirtualChannel.8.Main.PV |
| VirtualChannel.8.Alarm1.Type | As VirtualChannel1.Alarm1.Type | uint8 | 1fc0 | 8128 | Not applicable |
| VirtualChannel.8.Alarm2.Acknowledge | 1 = acknowledge alarm | bool | 01cf | 463 | Not applicable |
| VirtualChannel.8.Alarm2.Acknowledgement | 1 = alarm acknowledged | bool | 1ff0 | 8176 | Not applicable |
| VirtualChannel.8.Alarm2.Active | 1 = alarm source active, or safe but not ack'd | bool | 1feb | 8171 | Not applicable |
| VirtualChannel.8.Alarm2.Amount | Rate-of-change alarm 'Amount' | float32 | 1fe8 | 8168 | Same as VirtualChannel.8.Main.PV |
| VirtualChannel.8.Alarm2.AverageTime | Rate-of-change alarm 'Average time' | time_t | 1fea | 8170 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.8.Alarm2.Block | 0 = Blocking alarms off; 1 = Blocking alarms on | uint8 | 1fe2 | 8162 | Not applicable |
| VirtualChannel.8.Alarm2.ChangeTime | Rate-of-change alarm 'Change Time' | uint8 | 1fe9 | 8169 | Not applicable |
| VirtualChannel.8.Alarm2.Deviation | Deviation alarm 'Deviation Value' | float32 | 1fe7 | 8167 | Same as VirtualChannel.8.Main.PV |
| VirtualChannel.8.Alarm2.Dwell | Alarm dwell time | time_t | 1fe5 | 8165 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.8.Alarm2.Hysteresis | Alarm hysteresis value | float32 | 1fe4 | 8164 | Same as VirtualChannel.8.Main.PV |
| VirtualChannel.8.Alarm2.Inactive | 1 = alarm source safe and ack'd (if necessary) | bool | 1fee | 8174 | Not applicable |
| VirtualChannel.8.Alarm2.Inhibit | 1 = alarm inhibited | bool | 1ff1 | 8177 | Not applicable |
| VirtualChannel.8.Alarm2.Latch | As VirtualChannel1.Alarm1.Latch | uint8 | 1fe1 | 8161 | Not applicable |
| VirtualChannel.8.Alarm2.NotAcknowledged | 1 = alarm has not been acknowledged | bool | 1fef | 8175 | Not applicable |
| VirtualChannel.8.Alarm2.Reference | Deviation alarm 'Reference' value | float32 | 1fe6 | 8166 | Same as VirtualChannel.8.Main.PV |
| VirtualChannel.8.Alarm2.Status | As VirtualChannel1.Alarm1.Status | uint8 | 013f | 319 | Not applicable |
| VirtualChannel.8.Alarm2.Threshold | Alarm trigger threshold | float32 | 1fe3 | 8163 | Same as VirtualChannel.8.Main.PV |
| VirtualChannel.8.Alarm2.Type | As VirtualChannel1.Alarm1.Type | uint8 | 1fe0 | 8160 | Not applicable |
| VirtualChannel.8.Main.Descriptor | Virtual Channel descriptor | string_t | 4bbd | 19389 | Not applicable |
| VirtualChannel.8.Main.Disable | 1 = Virtual channel disabled | bool | 1fa3 | 8099 | Not applicable |
| VirtualChannel.8.Main.HighCutOff | The highest input value that will be totalised/counted | float32 | 1f85 | 8069 | Set by VirtualChannel.8.Main.Resolution |
| VirtualChannel.8.Main.Input1 | Input 1 value | float32 | 1f87 | 8071 | Set by VirtualChannel.8.Main.Resolution |
| VirtualChannel.8.Main.Input2 | Input 2 value | float32 | 1f88 | 8072 | Set by VirtualChannel.8.Main.Resolution |
| VirtualChannel.8.Main.LowCutOff | The lowest input value that will be totalised/counted | float32 | 1f84 | 8068 | Set by VirtualChannel.8.Main.Resolution |
| VirtualChannel.8.Main.ModbusInput | Modbus input value | float32 | 1f86 | 8070 | Set by VirtualChannel.8.Main.Resolution |
| VirtualChannel.8.Main.Operation | As VirtualChannel1.Main.Operation | uint8 | 1f81 | 8065 | Not applicable |
| VirtualChannel.8.Main.Period | The time period over which the calculation is made | int32 | 1f8a | 8074 | Not applicable |
| VirtualChannel.8.Main.Preset | Initiate preset. 0 = No; 1 = Yes | bool | 1f8c | 8076 | Not applicable |
| VirtualChannel.8.Main.PresetValue | The Preset value | float32 | 1f8d | 8077 | Set by VirtualChannel.8.Main.Resolution |
| VirtualChannel.8.Main.PV | The virtual channel output value | float32 | 013c | 316 | Set by VirtualChannel.8.Main.Resolution |
| VirtualChannel.8.Main.Reset | Initiate reset. 0 = No; 1 = Yes | bool | 1f8b | 8075 | Not applicable |
| VirtualChannel.8.Main.Resolution | Number of decimal places (0 to 6) | uint8 | 1f82 | 8066 | Not applicable |
| VirtualChannel.8.Main.Rollover | A pulse signal to indicate PV (output) has just rolled over | bool | 1f91 | 8081 | Not applicable |
| VirtualChannel.8.Main.RolloverValue | Rollover value | float32 | 1f92 | 8082 | Set by VirtualChannel.8.Main.Resolution |
| VirtualChannel.8.Main.Status | As VirtualChannel1.Main.Status | uint8 | 013d | 317 | Not applicable |
| VirtualChannel.8.Main.TimeRemaining | Time remaining before the calculation is made | time_t | 1f89 | 8073 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.8.Main.Trigger | Increment/decrement counter. 0 = No; 1 = Yes | bool | 1f8e | 8078 | Not applicable |
| VirtualChannel.8.Main.Type | As VirtualChannel1.Main.Type | uint8 | 1f80 | 8064 | Not applicable |
| VirtualChannel.8.Main.Units | Units descriptor | string_t | 4bd2 | 19410 | Not applicable |
| VirtualChannel.8.Main.UnitsScaler | Units scaler for totalisers | float32 | 1f83 | 8067 | 1dp |
| VirtualChannel.8.Trend.Colour | As VirtualChannel1.Trend.Colour | uint8 | 1fa0 | 8096 | Not applicable |
| VirtualChannel.8.Trend.SpanHigh | Specifies the highest PV (output value) to be displayed | float32 | 1fa2 | 8098 | Same as VirtualChannel.8.Main.PV |
| VirtualChannel.8.Trend.SpanLow | Specifies the lowest PV (output value) to be displayed | float32 | 1fa1 | 8097 | Same as VirtualChannel.8.Main.PV |
| VirtualChannel.9.Alarm1.Acknowledge | 1 = acknowledge alarm | bool | 01d0 | 464 | Not applicable |
| VirtualChannel.9.Alarm1.Acknowledgement | 1 = alarm acknowledged | bool | 2050 | 8272 | Not applicable |
| VirtualChannel.9.Alarm1.Active | 1 = alarm source active, or safe but not ack'd | bool | 204b | 8267 | Not applicable |
| VirtualChannel.9.Alarm1.Amount | Rate-of-change alarm 'Amount' | float32 | 2048 | 8264 | Same as VirtualChannel.9.Main.PV |
| VirtualChannel.9.Alarm1.AverageTime | Rate-of-change alarm 'Average time' | time_t | 204a | 8266 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.9.Alarm1.Block | 0 = Blocking alarms off; 1 = Blocking alarms on | uint8 | 2042 | 8258 | Not applicable |
| VirtualChannel.9.Alarm1.ChangeTime | Rate-of-change alarm 'Change Time' | uint8 | 2049 | 8265 | Not applicable |
| VirtualChannel.9.Alarm1.Deviation | Deviation alarm 'Deviation Value' | float32 | 2047 | 8263 | Same as VirtualChannel.9.Main.PV |
| VirtualChannel.9.Alarm1.Dwell | Alarm dwell time | time_t | 2045 | 8261 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.9.Alarm1.Hysteresis | Alarm hysteresis value | float32 | 2044 | 8260 | Same as VirtualChannel.9.Main.PV |
| VirtualChannel.9.Alarm1.Inactive | 1 = alarm source safe and ack'd (if necessary) | bool | 204e | 8270 | Not applicable |
| VirtualChannel.9.Alarm1.Inhibit | 1 = alarm inhibited | bool | 2051 | 8273 | Not applicable |
| VirtualChannel.9.Alarm1.Latch | As VirtualChannel1.Alarm1.Latch | uint8 | 2041 | 8257 | Not applicable |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| VirtualChannel.9.Alarm1.NotAcknowledged | 1 = alarm has not been acknowledged | bool | 204f | 8271 | Not applicable |
| VirtualChannel.9.Alarm1.Reference | Deviation alarm 'Reference' value | float32 | 2046 | 8262 | Same as VirtualChannel.9.Main.PV |
| VirtualChannel.9.Alarm1.Status | As VirtualChannel1.Alarm1.Status | uint8 | 0142 | 322 | Not applicable |
| VirtualChannel.9.Alarm1.Threshold | Alarm trigger threshold | float32 | 2043 | 8259 | Same as VirtualChannel.9.Main.PV |
| VirtualChannel.9.Alarm1.Type | As VirtualChannel1.Alarm1.Type | uint8 | 2040 | 8256 | Not applicable |
| VirtualChannel.9.Alarm2.Acknowledge | 1 = acknowledge alarm | bool | 01d1 | 465 | Not applicable |
| VirtualChannel.9.Alarm2.Acknowledgement | 1 = alarm acknowledged | bool | 2070 | 8304 | Not applicable |
| VirtualChannel.9.Alarm2.Active | 1 = alarm source active, or safe but not ack'd | bool | 206b | 8299 | Not applicable |
| VirtualChannel.9.Alarm2.Amount | Rate-of-change alarm 'Amount't | float32 | 2068 | 8296 | Same as VirtualChannel.9.Main.PV |
| VirtualChannel.9.Alarm2.AverageTime | Rate-of-change alarm 'Average time' | time_t | 206a | 8298 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.9.Alarm2.Block | 0 = Blocking alarms off; 1 = Blocking alarms on | uint8 | 2062 | 8290 | Not applicable |
| VirtualChannel.9.Alarm2.ChangeTime | Rate-of-change alarm 'Change Time' | uint8 | 2069 | 8297 | Not applicable |
| VirtualChannel.9.Alarm2.Deviation | Deviation alarm 'Deviation Value' | float32 | 2067 | 8295 | Same as VirtualChannel.9.Main.PV |
| VirtualChannel.9.Alarm2.Dwell | Alarm dwell time | time_t | 2065 | 8293 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.9.Alarm2.Hysteresis | Alarm hysteresis value | float32 | 2064 | 8292 | Same as VirtualChannel.9.Main.PV |
| VirtualChannel.9.Alarm2.Inactive | 1 = alarm source safe and ack'd (if necessary) | bool | 206e | 8302 | Not applicable |
| VirtualChannel.9.Alarm2.Inhibit | Inhibit | bool | 2071 | 8305 | Not applicable |
| VirtualChannel.9.Alarm2.Latch | As VirtualChannel1.Alarm1.Latch | uint8 | 2061 | 8289 | Not applicable |
| VirtualChannel.9.Alarm2.NotAcknowledged | 1 = alarm has not been acknowledged | bool | 206f | 8303 | Not applicable |
| VirtualChannel.9.Alarm2.Reference | Deviation alarm 'Reference' value | float32 | 2066 | 8294 | Same as VirtualChannel.9.Main.PV |
| VirtualChannel.9.Alarm2.Status | As VirtualChannel1.Alarm1.Status | uint8 | 0143 | 323 | Not applicable |
| VirtualChannel.9.Alarm2.Threshold | Alarm trigger threshold | float32 | 2063 | 8291 | Same as VirtualChannel.9.Main.PV |
| VirtualChannel.9.Alarm2.Type | As VirtualChannel1.Alarm1.Type | uint8 | 2060 | 8288 | Not applicable |
| VirtualChannel.9.Main.Descriptor | Virtual Channel descriptor | string_t | 4bd8 | 19416 | Not applicable |
| VirtualChannel.9.Main.Disable | 1 = Virtual channel disabled | bool | 2023 | 8227 | Not applicable |
| VirtualChannel.9.Main.HighCutOff | The highest input value that will be totalised/counted | float32 | 2005 | 8197 | Set by VirtualChannel.9.Main.Resolution |
| VirtualChannel.9.Main.Input1 | Input 1 value | float32 | 2007 | 8199 | Set by VirtualChannel.9.Main.Resolution |
| VirtualChannel.9.Main.Input2 | Input 2 value | float32 | 2008 | 8200 | Set by VirtualChannel.9.Main.Resolution |
| VirtualChannel.9.Main.LowCutOff | The lowest input value that will be totalised/counted | float32 | 2004 | 8196 | Set by VirtualChannel.9.Main.Resolution |
| VirtualChannel.9.Main.ModbusInput | Modbus input value | float32 | 2006 | 8198 | Set by VirtualChannel.9.Main.Resolution |
| VirtualChannel.9.Main.Operation | As VirtualChannel1.Main.Operation | uint8 | 2001 | 8193 | Not applicable |
| VirtualChannel.9.Main.Period | The time period over which the calculation is made | int32 | 200a | 8202 | Not applicable |
| VirtualChannel.9.Main.Preset | Initiate preset. 0 = No; 1 = Yes | bool | 200c | 8204 | Not applicable |
| VirtualChannel.9.Main.PresetValue | The Preset value | float32 | 200d | 8205 | Set by VirtualChannel.9.Main.Resolution |
| VirtualChannel.9.Main.PV | The virtual channel output value | float32 | 0140 | 320 | Set by VirtualChannel.9.Main.Resolution |
| VirtualChannel.9.Main.Reset | Initiate reset. 0 = No; 1 = Yes | bool | 200b | 8203 | Not applicable |
| VirtualChannel.9.Main.Resolution | Number of decimal places (0 to 6) | uint8 | 2002 | 8194 | Not applicable |
| VirtualChannel.9.Main.Rollover | A pulse signal to indicate PV (output) has just rolled over | bool | 2011 | 8209 | Not applicable |
| VirtualChannel.9.Main.RolloverValue | Rollover value | float32 | 2012 | 8210 | Set by VirtualChannel.9.Main.Resolution |
| VirtualChannel.9.Main.Status | As VirtualChannel1.Main.Status | uint8 | 0141 | 321 | Not applicable |
| VirtualChannel.9.Main.TimeRemaining | Time remaining before the calculation is made | time_t | 2009 | 8201 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.9.Main.Trigger | Increment/decrement counter. 0 = No; 1 = Yes | bool | 200e | 8206 | Not applicable |
| VirtualChannel.9.Main.Type | As VirtualChannel1.Main.Type | uint8 | 2000 | 8192 | Not applicable |
| VirtualChannel.9.Main.Units | Units descriptor | string_t | 4bed | 19437 | Not applicable |
| VirtualChannel.9.Main.UnitsScaler | Units scaler for totalisers | float32 | 2003 | 8195 | 1dp |
| VirtualChannel.9.Trend.Colour | As VirtualChannel1.Trend.Colour | uint8 | 2020 | 8224 | Not applicable |
| VirtualChannel.9.Trend.SpanHigh | Specifies the highest PV (output value) to be displayed | float32 | 2022 | 8226 | Same as VirtualChannel.9.Main.PV |
| VirtualChannel.9.Trend.SpanLow | Specifies the lowest PV (output value) to be displayed | float32 | 2021 | 8225 | Same as VirtualChannel.9.Main.PV |
| VirtualChannel.10.Alarm1.Acknowledge | 1 = acknowledge alarm | bool | 01d2 | 466 | Not applicable |
| VirtualChannel.10.Alarm1.Acknowledgement | 1 = alarm acknowledged | bool | 20d0 | 8400 | Not applicable |
| VirtualChannel.10.Alarm1.Active | 1 = alarm source active, or safe but not ack'd | bool | 20cb | 8395 | Not applicable |
| VirtualChannel.10.Alarm1.Amount | Rate-of-change alarm 'Amount' | float32 | 20c8 | 8392 | Same as VirtualChannel.10.Main.PV |
| VirtualChannel.10.Alarm1.AverageTime | Rate-of-change alarm 'Average time' | time_t | 20ca | 8394 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.10.Alarm1.Block | 0 = Blocking alarms off; 1 = Blocking alarms on | uint8 | 20c2 | 8386 | Not applicable |
| VirtualChannel.10.Alarm1.ChangeTime | Rate-of-change alarm 'Change Time' | uint8 | 20c9 | 8393 | Not applicable |
| VirtualChannel.10.Alarm1.Deviation | Deviation alarm 'Deviation Value' | float32 | 20c7 | 8391 | Same as VirtualChannel.10.Main.PV |
| VirtualChannel.10.Alarm1.Dwell | Alarm dwell time | time_t | 20c5 | 8389 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.10.Alarm1.Hysteresis | Alarm hysteresis value | float32 | 20c4 | 8388 | Same as VirtualChannel.10.Main.PV |
| VirtualChannel.10.Alarm1.Inactive | 1 = alarm source safe and ack'd (if necessary) | bool | 20ce | 8398 | Not applicable |
| VirtualChannel.10.Alarm1.Inhibit | 1 = alarm inhibited | bool | 20d1 | 8401 | Not applicable |
| VirtualChannel.10.Alarm1.Latch | As VirtualChannel1.Alarm1.Latch | uint8 | 20c1 | 8385 | Not applicable |
| VirtualChannel.10.Alarm1.NotAcknowledged | 1 = alarm has not been acknowledged | bool | 20cf | 8399 | Not applicable |
| VirtualChannel.10.Alarm1.Reference | Deviation alarm 'Reference' value | float32 | 20c6 | 8390 | Same as VirtualChannel.10.Main.PV |
| VirtualChannel.10.Alarm1.Status | As VirtualChannel1.Alarm1.Status | uint8 | 0146 | 326 | Not applicable |
| VirtualChannel.10.Alarm1.Threshold | Alarm trigger threshold | float32 | 20c3 | 8387 | Same as VirtualChannel.10.Main.PV |
| VirtualChannel.10.Alarm1.Type | As VirtualChannel1.Alarm1.Type | uint8 | 20c0 | 8384 | Not applicable |
| VirtualChannel.10.Alarm2.Acknowledge | 1 = acknowledge alarm | bool | 01d3 | 467 | Not applicable |
| VirtualChannel.10.Alarm2.Acknowledgement | 1 = alarm acknowledged | bool | 20f0 | 8432 | Not applicable |
| VirtualChannel.10.Alarm2.Active | 1 = alarm source active, or safe but not ack'd | bool | 20eb | 8427 | Not applicable |
| VirtualChannel.10.Alarm2.Amount | Rate-of-change alarm 'Amount' | float32 | 20e8 | 8424 | Same as VirtualChannel.10.Main.PV |
| VirtualChannel.10.Alarm2.AverageTime | Rate-of-change alarm 'Average time' | time_t | 20ea | 8426 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.10.Alarm2.Block | 0 = Blocking alarms off; 1 = Blocking alarms on | uint8 | 20e2 | 8418 | Not applicable |
| VirtualChannel.10.Alarm2.ChangeTime | Rate-of-change alarm 'Change Time' | uint8 | 20e9 | 8425 | Not applicable |
| VirtualChannel.10.Alarm2.Deviation | Deviation alarm 'Deviation Value' | float32 | 20e7 | 8423 | Same as VirtualChannel.10.Main.PV |
| VirtualChannel.10.Alarm2.Dwell | Alarm dwell time | time_t | 20e5 | 8421 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.10.Alarm2.Hysteresis | Alarm hysteresis value | float32 | 20e4 | 8420 | Same as VirtualChannel.10.Main.PV |
| VirtualChannel.10.Alarm2.Inactive | 1 = alarm source safe and ack'd (if necessary) | bool | 20ee | 8430 | Not applicable |
| VirtualChannel.10.Alarm2.Inhibit | 1 = alarm inhibited | bool | 20f1 | 8433 | Not applicable |
| VirtualChannel.10.Alarm2.Latch | As VirtualChannel1.Alarm1.Latch | uint8 | 20e1 | 8417 | Not applicable |
| VirtualChannel.10.Alarm2.NotAcknowledged | 1 = alarm has not been acknowledged | bool | 20ef | 8431 | Not applicable |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| VirtualChannel.10.Alarm2.Reference | Deviation alarm 'Reference' value | float32 | 20e6 | 8422 | Same as VirtualChannel.10.Main.PV |
| VirtualChannel.10.Alarm2.Status | As VirtualChannel1.Alarm1.Status | uint8 | 0147 | 327 | Not applicable |
| VirtualChannel.10.Alarm2.Threshold | Alarm trigger threshold | float32 | 20e3 | 8419 | Same as VirtualChannel.10.Main.PV |
| VirtualChannel.10.Alarm2.Type | As VirtualChannel1.Alarm1.Type | uint8 | 20e0 | 8416 | Not applicable |
| VirtualChannel.10.Main.Descriptor | Virtual Channel descriptor | string_t | 4bf3 | 19443 | Not applicable |
| VirtualChannel.10.Main.Disable | 1 = Virtual channel disabled | bool | 20a3 | 8355 | Not applicable |
| VirtualChannel.10.Main.HighCutOff | The highest input value that will be totalised/counted | float32 | 2085 | 8325 | Set by VirtualChannel.10.Main.Resolution |
| VirtualChannel.10.Main.Input1 | Input 1 value | float32 | 2087 | 8327 | Set by VirtualChannel.10.Main.Resolution |
| VirtualChannel.10.Main.Input2 | Input 2 value | float32 | 2088 | 8328 | Set by VirtualChannel.10.Main.Resolution |
| VirtualChannel.10.Main.LowCutOff | The lowest input value that will be totalised/counted | float32 | 2084 | 8324 | Set by VirtualChannel.10.Main.Resolution |
| VirtualChannel.10.Main.ModbusInput | Modbus input value | float32 | 2086 | 8326 | Set by VirtualChannel.10.Main.Resolution |
| VirtualChannel.10.Main.Operation | As VirtualChannel1.Main.Operation | uint8 | 2081 | 8321 | Not applicable |
| VirtualChannel.10.Main.Period | Averaging period | int32 | 208a | 8330 | Not applicable |
| VirtualChannel.10.Main.Preset | Initiate preset. 0 = No; 1 = Yes | bool | 208c | 8332 | Not applicable |
| VirtualChannel.10.Main.PresetValue | The Preset value | float32 | 208d | 8333 | Set by VirtualChannel.10.Main.Resolution |
| VirtualChannel.10.Main.PV | The virtual channel output value | float32 | 0144 | 324 | Set by VirtualChannel.10.Main.Resolution |
| VirtualChannel.10.Main.Reset | Initiate reset. 0 = No; 1 = Yes | bool | 208b | 8331 | Not applicable |
| VirtualChannel.10.Main.Resolution | Number of decimal places (0 to 6) | uint8 | 2082 | 8322 | Not applicable |
| VirtualChannel.10.Main.Rollover | A pulse signal to indicate PV (output) has just rolled over | bool | 2091 | 8337 | Not applicable |
| VirtualChannel.10.Main.RolloverValue | Rollover value | float32 | 2092 | 8338 | Set by VirtualChannel.10.Main.Resolution |
| VirtualChannel.10.Main.Status | As VirtualChannel1.Main.Status | uint8 | 0145 | 325 | Not applicable |
| VirtualChannel.10.Main.TimeRemaining | Time remaining before the calculation is made | time_t | 2089 | 8329 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.10.Main.Trigger | Increment/decrement counter. 0 = No; 1 = Yes | bool | 208e | 8334 | Not applicable |
| VirtualChannel.10.Main.Type | As VirtualChannel1.Main.Type | uint8 | 2080 | 8320 | Not applicable |
| VirtualChannel.10.Main.Units | Units descriptor | string_t | 4c08 | 19464 | Not applicable |
| VirtualChannel.10.Main.UnitsScaler | Units scaler for totalisers | float32 | 2083 | 8323 | 1dp |
| VirtualChannel.10.Trend.Colour | As VirtualChannel1.Trend.Colour | uint8 | 20a0 | 8352 | Not applicable |
| VirtualChannel.10.Trend.SpanHigh | Specifies the highest PV (output value) to be displayed | float32 | 20a2 | 8354 | Same as VirtualChannel.10.Main.PV |
| VirtualChannel.10.Trend.SpanLow | Specifies the lowest PV (output value) to be displayed | float32 | 20a1 | 8353 | Same as VirtualChannel.10.Main.PV |
| | | | | | |
| VirtualChannel.11.Alarm1.Acknowledge | 1 = acknowledge alarm | bool | 01d4 | 468 | Not applicable |
| VirtualChannel.11.Alarm1.Acknowledgement | 1 = alarm acknowledged | bool | 2150 | 8528 | Not applicable |
| VirtualChannel.11.Alarm1.Active | 1 = alarm source active, or safe but not ack'd | bool | 214b | 8523 | Not applicable |
| VirtualChannel.11.Alarm1.Amount | Rate-of-change alarm 'Amount' | float32 | 2148 | 8520 | Same as VirtualChannel.11.Main.PV |
| VirtualChannel.11.Alarm1.AverageTime | Rate-of-change alarm 'Average time' | time_t | 214a | 8522 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.11.Alarm1.Block | 0 = Blocking alarms off; 1 = Blocking alarms on | uint8 | 2142 | 8514 | Not applicable |
| VirtualChannel.11.Alarm1.ChangeTime | Rate-of-change alarm 'Change Time' | uint8 | 2149 | 8521 | Not applicable |
| VirtualChannel.11.Alarm1.Deviation | Deviation alarm 'Deviation Value' | float32 | 2147 | 8519 | Same as VirtualChannel.11.Main.PV |
| VirtualChannel.11.Alarm1.Dwell | Alarm dwell time | time_t | 2145 | 8517 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.11.Alarm1.Hysteresis | Alarm hysteresis value | float32 | 2144 | 8516 | Same as VirtualChannel.11.Main.PV |
| VirtualChannel.11.Alarm1.Inactive | 1 = alarm source safe and ack'd (if necessary) | bool | 214e | 8526 | Not applicable |
| VirtualChannel.11.Alarm1.Inhibit | 1 = alarm inhibited | bool | 2151 | 8529 | Not applicable |
| VirtualChannel.11.Alarm1.Latch | As VirtualChannel1.Alarm1.Latch | uint8 | 2141 | 8513 | Not applicable |
| VirtualChannel.11.Alarm1.NotAcknowledged | 1 = alarm has not been acknowledged | bool | 214f | 8527 | Not applicable |
| VirtualChannel.11.Alarm1.Reference | Deviation alarm 'Reference' value | float32 | 2146 | 8518 | Same as VirtualChannel.11.Main.PV |
| VirtualChannel.11.Alarm1.Status | As VirtualChannel1.Alarm1.Status | uint8 | 014a | 330 | Not applicable |
| VirtualChannel.11.Alarm1.Threshold | Alarm trigger threshold | float32 | 2143 | 8515 | Same as VirtualChannel.11.Main.PV |
| VirtualChannel.11.Alarm1.Type | As VirtualChannel1.Alarm1.Type | uint8 | 2140 | 8512 | Not applicable |
| VirtualChannel.11.Alarm2.Acknowledge | 1 = acknowledge alarm | bool | 01d5 | 469 | Not applicable |
| VirtualChannel.11.Alarm2.Acknowledgement | 1 = alarm acknowledged | bool | 2170 | 8560 | Not applicable |
| VirtualChannel.11.Alarm2.Active | 1 = alarm source active, or safe but not ack'd | bool | 216b | 8555 | Not applicable |
| VirtualChannel.11.Alarm2.Amount | Rate-of-change alarm 'Amount' | float32 | 2168 | 8552 | Same as VirtualChannel.11.Main.PV |
| VirtualChannel.11.Alarm2.AverageTime | Rate-of-change alarm 'Average time' | time_t | 216a | 8554 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.11.Alarm2.Block | 0 = Blocking alarms off; 1 = Blocking alarms on | uint8 | 2162 | 8546 | Not applicable |
| VirtualChannel.11.Alarm2.ChangeTime | Rate-of-change alarm 'Change Time' | uint8 | 2169 | 8553 | Not applicable |
| VirtualChannel.11.Alarm2.Deviation | Deviation alarm 'Deviation Value' | float32 | 2167 | 8551 | Same as VirtualChannel.11.Main.PV |
| VirtualChannel.11.Alarm2.Dwell | Alarm dwell time | time_t | 2165 | 8549 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.11.Alarm2.Hysteresis | Alarm hysteresis value | float32 | 2164 | 8548 | Same as VirtualChannel.11.Main.PV |
| VirtualChannel.11.Alarm2.Inactive | 1 = alarm source safe and ack'd (if necessary) | bool | 216e | 8558 | Not applicable |
| VirtualChannel.11.Alarm2.Inhibit | 1 = alarm inhibited | bool | 2171 | 8561 | Not applicable |
| VirtualChannel.11.Alarm2.Latch | As VirtualChannel1.Alarm1.Latch | uint8 | 2161 | 8545 | Not applicable |
| VirtualChannel.11.Alarm2.NotAcknowledged | 1 = alarm has not been acknowledged | bool | 216f | 8559 | Not applicable |
| VirtualChannel.11.Alarm2.Reference | Deviation alarm 'Reference' value | float32 | 2166 | 8550 | Same as VirtualChannel.11.Main.PV |
| VirtualChannel.11.Alarm2.Status | As VirtualChannel1.Alarm1.Status | uint8 | 014b | 331 | Not applicable |
| VirtualChannel.11.Alarm2.Threshold | Alarm trigger threshold | float32 | 2163 | 8547 | Same as VirtualChannel.11.Main.PV |
| VirtualChannel.11.Alarm2.Type | As VirtualChannel1.Alarm1.Type | uint8 | 2160 | 8544 | Not applicable |
| VirtualChannel.11.Main.Descriptor | Virtual Channel descriptor | string_t | 4c0e | 19470 | Not applicable |
| VirtualChannel.11.Main.Disable | 1 = Virtual channel disabled | bool | 2123 | 8483 | Not applicable |
| VirtualChannel.11.Main.HighCutOff | The highest input value that will be totalised/counted | float32 | 2105 | 8453 | Set by VirtualChannel.11.Main.Resolution |
| VirtualChannel.11.Main.Input1 | Input 1 value | float32 | 2107 | 8455 | Set by VirtualChannel.11.Main.Resolution |
| VirtualChannel.11.Main.Input2 | Input 2 value | float32 | 2108 | 8456 | Set by VirtualChannel.11.Main.Resolution |
| VirtualChannel.11.Main.LowCutOff | The lowest input value that will be totalised/counted | float32 | 2104 | 8452 | Set by VirtualChannel.11.Main.Resolution |
| VirtualChannel.11.Main.ModbusInput | Modbus input value | float32 | 2106 | 8454 | Set by VirtualChannel.11.Main.Resolution |
| VirtualChannel.11.Main.Operation | As VirtualChannel1.Main.Operation | uint8 | 2101 | 8449 | Not applicable |
| VirtualChannel.11.Main.Period | The time period over which the calculation is made | int32 | 210a | 8458 | Not applicable |
| VirtualChannel.11.Main.Preset | Initiate preset. 0 = No; 1 = Yes | bool | 210c | 8460 | Not applicable |
| VirtualChannel.11.Main.PresetValue | The Preset value | float32 | 210d | 8461 | Set by VirtualChannel.11.Main.Resolution |
| VirtualChannel.11.Main.PV | The virtual channel output value | float32 | 0148 | 328 | Set by VirtualChannel.11.Main.Resolution |
| VirtualChannel.11.Main.Reset | Initiate reset. 0 = No; 1 = Yes | bool | 210b | 8459 | Not applicable |
| VirtualChannel.11.Main.Resolution | Number of decimal places (0 to 6) | uint8 | 2102 | 8450 | Not applicable |
| VirtualChannel.11.Main.Rollover | A pulse signal to indicate PV (output) has just rolled over | bool | 2111 | 8465 | Not applicable |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| VirtualChannel.11.Main.RolloverValue | Rollover value | float32 | 2112 | 8466 | Set by VirtualChannel.11.Main.Resolution |
| VirtualChannel.11.Main.Status | As VirtualChannel1.Main.Status | uint8 | 0149 | 329 | Not applicable |
| VirtualChannel.11.Main.TimeRemaining | Time remaining before the calculation is made | time_t | 2109 | 8457 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.11.Main.Trigger | Increment/decrement counter. 0 = No; 1 = Yes | bool | 210e | 8462 | Not applicable |
| VirtualChannel.11.Main.Type | As VirtualChannel1.Main.Type | uint8 | 2100 | 8448 | Not applicable |
| VirtualChannel.11.Main.Units | Units descriptor | string_t | 4c23 | 19491 | Not applicable |
| VirtualChannel.11.Main.UnitsScaler | Units scaler for totalisers | float32 | 2103 | 8451 | 1dp |
| VirtualChannel.11.Trend.Colour | As VirtualChannel1.Trend.Colour | uint8 | 2120 | 8480 | Not applicable |
| VirtualChannel.11.Trend.SpanHigh | Specifies the highest PV (output value) to be displayed | float32 | 2122 | 8482 | Same as VirtualChannel.11.Main.PV |
| VirtualChannel.11.Trend.SpanLow | Specifies the lowest PV (output value) to be displayed | float32 | 2121 | 8481 | Same as VirtualChannel.11.Main.PV |
| | | | | | |
| VirtualChannel.12.Alarm1.Acknowledge | 1 = acknowledge alarm | bool | 01d6 | 470 | Not applicable |
| VirtualChannel.12.Alarm1.Acknowledgement | 1 = alarm acknowledged | bool | 21d0 | 8656 | Not applicable |
| VirtualChannel.12.Alarm1.Active | 1 = alarm source active, or safe but not ack'd | bool | 21cb | 8651 | Not applicable |
| VirtualChannel.12.Alarm1.Amount | Rate-of-change alarm 'Amount' | float32 | 21c8 | 8648 | Same as VirtualChannel.12.Main.PV |
| VirtualChannel.12.Alarm1.AverageTime | Rate-of-change alarm 'Average time' | time_t | 21ca | 8650 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.12.Alarm1.Block | 0 = Blocking alarms off; 1 = Blocking alarms on | uint8 | 21c2 | 8642 | Not applicable |
| VirtualChannel.12.Alarm1.ChangeTime | Rate-of-change alarm 'Change Time' | uint8 | 21c9 | 8649 | Not applicable |
| VirtualChannel.12.Alarm1.Deviation | Deviation alarm 'Deviation Value' | float32 | 21c7 | 8647 | Same as VirtualChannel.12.Main.PV |
| VirtualChannel.12.Alarm1.Dwell | Alarm dwell time | time_t | 21c5 | 8645 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.12.Alarm1.Hysteresis | Alarm hysteresis value | float32 | 21c4 | 8644 | Same as VirtualChannel.12.Main.PV |
| VirtualChannel.12.Alarm1.Inactive | 1 = alarm source safe and ack'd (if necessary) | bool | 21ce | 8654 | Not applicable |
| VirtualChannel.12.Alarm1.Inhibit | 1 = alarm inhibited | bool | 21d1 | 8657 | Not applicable |
| VirtualChannel.12.Alarm1.Latch | As VirtualChannel1.Alarm1.Latch | uint8 | 21c1 | 8641 | Not applicable |
| VirtualChannel.12.Alarm1.NotAcknowledged | 1 = alarm has not been acknowledged | bool | 21cf | 8655 | Not applicable |
| VirtualChannel.12.Alarm1.Reference | Deviation alarm 'Reference' value | float32 | 21c6 | 8646 | Same as VirtualChannel.12.Main.PV |
| VirtualChannel.12.Alarm1.Status | As VirtualChannel1.Alarm1.Status | uint8 | 014e | 334 | Not applicable |
| VirtualChannel.12.Alarm1.Threshold | Alarm trigger threshold | float32 | 21c3 | 8643 | Same as VirtualChannel.12.Main.PV |
| VirtualChannel.12.Alarm1.Type | As VirtualChannel1.Alarm1.Type | uint8 | 21c0 | 8640 | Not applicable |
| VirtualChannel.12.Alarm2.Acknowledge | 1 = acknowledge alarm | bool | 01d7 | 471 | Not applicable |
| VirtualChannel.12.Alarm2.Acknowledgement | 1 = alarm acknowledged | bool | 21f0 | 8688 | Not applicable |
| VirtualChannel.12.Alarm2.Active | 1 = alarm source active, or safe but not ack'd | bool | 21eb | 8683 | Not applicable |
| VirtualChannel.12.Alarm2.Amount | Rate-of-change alarm 'Amount' | float32 | 21e8 | 8680 | Same as VirtualChannel.12.Main.PV |
| VirtualChannel.12.Alarm2.AverageTime | Rate-of-change alarm 'Average time' | time_t | 21ea | 8682 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.12.Alarm2.Block | 0 = Blocking alarms off; 1 = Blocking alarms on | uint8 | 21e2 | 8674 | Not applicable |
| VirtualChannel.12.Alarm2.ChangeTime | Rate-of-change alarm 'Change Time' | uint8 | 21e9 | 8681 | Not applicable |
| VirtualChannel.12.Alarm2.Deviation | Deviation alarm 'Deviation Value' | float32 | 21e7 | 8679 | Same as VirtualChannel.12.Main.PV |
| VirtualChannel.12.Alarm2.Dwell | Alarm dwell time | time_t | 21e5 | 8677 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.12.Alarm2.Hysteresis | Alarm hysteresis value | float32 | 21e4 | 8676 | Same as VirtualChannel.12.Main.PV |
| VirtualChannel.12.Alarm2.Inactive | 1 = alarm source safe and ack'd (if necessary) | bool | 21ee | 8686 | Not applicable |
| VirtualChannel.12.Alarm2.Inhibit | 1 = alarm inhibited | bool | 21f1 | 8689 | Not applicable |
| VirtualChannel.12.Alarm2.Latch | As VirtualChannel1.Alarm1.Latch | uint8 | 21e1 | 8673 | Not applicable |
| VirtualChannel.12.Alarm2.NotAcknowledged | 1 = alarm has not been acknowledged | bool | 21ef | 8687 | Not applicable |
| VirtualChannel.12.Alarm2.Reference | Deviation alarm 'Reference' value | float32 | 21e6 | 8678 | Same as VirtualChannel.12.Main.PV |
| VirtualChannel.12.Alarm2.Status | As VirtualChannel1.Alarm1.Status | uint8 | 014f | 335 | Not applicable |
| VirtualChannel.12.Alarm2.Threshold | Alarm trigger threshold | float32 | 21e3 | 8675 | Same as VirtualChannel.12.Main.PV |
| VirtualChannel.12.Alarm2.Type | As VirtualChannel1.Alarm1.Type | uint8 | 21e0 | 8672 | Not applicable |
| VirtualChannel.12.Main.Descriptor | Virtual Channel descriptorl | string_t | 4c29 | 19497 | Not applicable |
| VirtualChannel.12.Main.Disable | 1 = Virtual channel disabled | bool | 21a3 | 8611 | Not applicable |
| VirtualChannel.12.Main.HighCutOff | The highest input value that will be totalised/counted | float32 | 2185 | 8581 | Set by VirtualChannel.12.Main.Resolution |
| VirtualChannel.12.Main.Input1 | Input 1 value | float32 | 2187 | 8583 | Set by VirtualChannel.12.Main.Resolution |
| VirtualChannel.12.Main.Input2 | Input 2 value | float32 | 2188 | 8584 | Set by VirtualChannel.12.Main.Resolution |
| VirtualChannel.12.Main.LowCutOff | The lowest input value that will be totalised/counted | float32 | 2184 | 8580 | Set by VirtualChannel.12.Main.Resolution |
| VirtualChannel.12.Main.ModbusInput | Modbus input value | float32 | 2186 | 8582 | Set by VirtualChannel.12.Main.Resolution |
| VirtualChannel.12.Main.Operation | As VirtualChannel1.Main.Operation | uint8 | 2181 | 8577 | Not applicable |
| VirtualChannel.12.Main.Period | The time period over which the calculation is made | int32 | 218a | 8586 | Not applicable |
| VirtualChannel.12.Main.Preset | Initiate preset. 0 = No; 1 = Yes | bool | 218c | 8588 | Not applicable |
| VirtualChannel.12.Main.PresetValue | The Preset value | float32 | 218d | 8589 | Set by VirtualChannel.12.Main.Resolution |
| VirtualChannel.12.Main.PV | The virtual channel output value | float32 | 014c | 332 | Set by VirtualChannel.12.Main.Resolution |
| VirtualChannel.12.Main.Reset | Initiate reset. 0 = No; 1 = Yes | bool | 218b | 8587 | Not applicable |
| VirtualChannel.12.Main.Resolution | Number of decimal places (0 to 6) | uint8 | 2182 | 8578 | Not applicable |
| VirtualChannel.12.Main.Rollover | A pulse signal to indicate PV (output) has just rolled over | bool | 2191 | 8593 | Not applicable |
| VirtualChannel.12.Main.RolloverValue | Rollover value | float32 | 2192 | 8594 | Set by VirtualChannel.12.Main.Resolution |
| VirtualChannel.12.Main.Status | As VirtualChannel1.Main.Status | uint8 | 014d | 333 | Not applicable |
| VirtualChannel.12.Main.TimeRemaining | Time remaining before the calculation is made | time_t | 2189 | 8585 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.12.Main.Trigger | Increment/decrement counter. 0 = No; 1 = Yes | bool | 218e | 8590 | Not applicable |
| VirtualChannel.12.Main.Type | As VirtualChannel1.Main.Type | uint8 | 2180 | 8576 | Not applicable |
| VirtualChannel.12.Main.Units | Units descriptor | string_t | 4c3e | 19518 | Not applicable |
| VirtualChannel.12.Main.UnitsScaler | Units scaler for totalisers | float32 | 2183 | 8579 | 1dp |
| VirtualChannel.12.Trend.Colour | As VirtualChannel1.Trend.Colour | uint8 | 21a0 | 8608 | Not applicable |
| VirtualChannel.12.Trend.SpanHigh | Specifies the highest PV (output value) to be displayed | float32 | 21a2 | 8610 | Same as VirtualChannel.12.Main.PV |
| VirtualChannel.12.Trend.SpanLow | Specifies the lowest PV (output value) to be displayed | float32 | 21a1 | 8609 | Same as VirtualChannel.12.Main.PV |
| | | | | | |
| VirtualChannel.13.Alarm1.Acknowledge | 1 = acknowledge alarm | bool | 01d8 | 472 | Not applicable |
| VirtualChannel.13.Alarm1.Acknowledgement | 1 = alarm acknowledged | bool | 2250 | 8784 | Not applicable |
| VirtualChannel.13.Alarm1.Active | 1 = alarm source active, or safe but not ack'd | bool | 224b | 8779 | Not applicable |
| VirtualChannel.13.Alarm1.Amount | Rate-of-change alarm 'Amount' | float32 | 2248 | 8776 | Same as VirtualChannel.13.Main.PV |
| VirtualChannel.13.Alarm1.AverageTime | Rate-of-change alarm 'Average time' | time_t | 224a | 8778 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.13.Alarm1.Block | 0 = Blocking alarms off; 1 = Blocking alarms on | uint8 | 2242 | 8770 | Not applicable |
| VirtualChannel.13.Alarm1.ChangeTime | Rate-of-change alarm 'Change Time' | uint8 | 2249 | 8777 | Not applicable |
| VirtualChannel.13.Alarm1.Deviation | Deviation alarm 'Deviation Value' | float32 | 2247 | 8775 | Same as VirtualChannel.13.Main.PV |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| VirtualChannel.13.Alarm1.Dwell | Alarm dwell time | time_t | 2245 | 8773 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.13.Alarm1.Hysteresis | Alarm hysteresis value | float32 | 2244 | 8772 | Same as VirtualChannel.13.Main.PV |
| VirtualChannel.13.Alarm1.Inhibit | 1 = alarm inhibited | bool | 2251 | 8785 | Not applicable |
| VirtualChannel.13.Alarm1.Inactive | 1 = alarm source safe and ack'd (if necessary) | bool | 224e | 8782 | Not applicable |
| VirtualChannel.13.Alarm1.Latch | As VirtualChannel1.Alarm1.Latch | uint8 | 2241 | 8769 | Not applicable |
| VirtualChannel.13.Alarm1.NotAcknowledged | 1 = alarm has not been acknowledged | bool | 224f | 8783 | Not applicable |
| VirtualChannel.13.Alarm1.Reference | Deviation alarm 'Reference' value | float32 | 2246 | 8774 | Same as VirtualChannel.13.Main.PV |
| VirtualChannel.13.Alarm1.Status | As VirtualChannel1.Alarm1.Status | uint8 | 0152 | 338 | Not applicable |
| VirtualChannel.13.Alarm1.Threshold | Alarm trigger threshold | float32 | 2243 | 8771 | Same as VirtualChannel.13.Main.PV |
| VirtualChannel.13.Alarm1.Type | As VirtualChannel1.Alarm1.Type | uint8 | 2240 | 8768 | Not applicable |
| VirtualChannel.13.Alarm2.Acknowledge | 1 = acknowledge alarm | bool | 01d9 | 473 | Not applicable |
| VirtualChannel.13.Alarm2.Acknowledgement | 1 = alarm acknowledged | bool | 2270 | 8816 | Not applicable |
| VirtualChannel.13.Alarm2.Active | 1 = alarm source active, or safe but not ack'd | bool | 226b | 8811 | Not applicable |
| VirtualChannel.13.Alarm2.Amount | Rate-of-change alarm 'Amount' | float32 | 2268 | 8808 | Same as VirtualChannel.13.Main.PV |
| VirtualChannel.13.Alarm2.AverageTime | Rate-of-change alarm 'Average time' | time_t | 226a | 8810 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.13.Alarm2.Block | 0 = Blocking alarms off; 1 = Blocking alarms on | uint8 | 2262 | 8802 | Not applicable |
| VirtualChannel.13.Alarm2.ChangeTime | Rate-of-change alarm 'Change Time' | uint8 | 2269 | 8809 | Not applicable |
| VirtualChannel.13.Alarm2.Deviation | Deviation alarm 'Deviation Value' | float32 | 2267 | 8807 | Same as VirtualChannel.13.Main.PV |
| VirtualChannel.13.Alarm2.Dwell | Alarm dwell time | time_t | 2265 | 8805 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.13.Alarm2.Hysteresis | Alarm hysteresis value | float32 | 2264 | 8804 | Same as VirtualChannel.13.Main.PV |
| VirtualChannel.13.Alarm2.Inactive | 1 = alarm source safe and ack'd (if necessary) | bool | 226e | 8814 | Not applicable |
| VirtualChannel.13.Alarm2.Inhibit | 1 = alarm inhibited | bool | 2271 | 8817 | Not applicable |
| VirtualChannel.13.Alarm2.Latch | As VirtualChannel1.Alarm1.Latch | uint8 | 2261 | 8801 | Not applicable |
| VirtualChannel.13.Alarm2.NotAcknowledged | 1 = alarm has not been acknowledged | bool | 226f | 8815 | Not applicable |
| VirtualChannel.13.Alarm2.Reference | Deviation alarm 'Reference' value | float32 | 2266 | 8806 | Same as VirtualChannel.13.Main.PV |
| VirtualChannel.13.Alarm2.Status | As VirtualChannel1.Alarm1.Status | uint8 | 0153 | 339 | Not applicable |
| VirtualChannel.13.Alarm2.Threshold | Alarm trigger threshold | float32 | 2263 | 8803 | Same as VirtualChannel.13.Main.PV |
| VirtualChannel.13.Alarm2.Type | As VirtualChannel1.Alarm1.Type | uint8 | 2260 | 8800 | Not applicable |
| VirtualChannel.13.Main.Descriptor | Virtual Channel descriptor | string_t | 4c44 | 19524 | Not applicable |
| VirtualChannel.13.Main.Disable | 1 = Virtual channel disabled | bool | 2223 | 8739 | Not applicable |
| VirtualChannel.13.Main.HighCutOff | The highest input value that will be totalised/counted | float32 | 2205 | 8709 | Set by VirtualChannel.13.Main.Resolution |
| VirtualChannel.13.Main.Input1 | Input 1 value | float32 | 2207 | 8711 | Set by VirtualChannel.13.Main.Resolution |
| VirtualChannel.13.Main.Input2 | Input 2 value | float32 | 2208 | 8712 | Set by VirtualChannel.13.Main.Resolution |
| VirtualChannel.13.Main.LowCutOff | The lowest input value that will be totalised/counted | float32 | 2204 | 8708 | Set by VirtualChannel.13.Main.Resolution |
| VirtualChannel.13.Main.ModbusInput | Modbus input value | float32 | 2206 | 8710 | Set by VirtualChannel.13.Main.Resolution |
| VirtualChannel.13.Main.Operation | As VirtualChannel1.Main.Operation | uint8 | 2201 | 8705 | Not applicable |
| VirtualChannel.13.Main.Period | The time period over which the calculation is made | int32 | 220a | 8714 | Not applicable |
| VirtualChannel.13.Main.Preset | Initiate preset. 0 = No; 1 = Yes | bool | 220c | 8716 | Not applicable |
| VirtualChannel.13.Main.PresetValue | The Preset value | float32 | 220d | 8717 | Set by VirtualChannel.13.Main.Resolution |
| VirtualChannel.13.Main.PV | The virtual channel output value | float32 | 0150 | 336 | Set by VirtualChannel.13.Main.Resolution |
| VirtualChannel.13.Main.Reset | Initiate reset. 0 = No; 1 = Yes | bool | 220b | 8715 | Not applicable |
| VirtualChannel.13.Main.Resolution | Number of decimal places (0 to 6) | uint8 | 2202 | 8706 | Not applicable |
| VirtualChannel.13.Main.Rollover | A pulse signal to indicate PV (output) has just rolled over | bool | 2211 | 8721 | Not applicable |
| VirtualChannel.13.Main.RolloverValue | Rollover value | float32 | 2212 | 8722 | Set by VirtualChannel.13.Main.Resolution |
| VirtualChannel.13.Main.Status | As VirtualChannel1.Main.Status | uint8 | 0151 | 337 | Not applicable |
| VirtualChannel.13.Main.TimeRemaining | Time remaining before the calculation is made | time_t | 2209 | 8713 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.13.Main.Trigger | Increment/decrement counter. 0 = No; 1 = Yes | bool | 220e | 8718 | Not applicable |
| VirtualChannel.13.Main.Type | As VirtualChannel1.Main.Type | uint8 | 2200 | 8704 | Not applicable |
| VirtualChannel.13.Main.Units | Units descriptor | string_t | 4c59 | 19545 | Not applicable |
| VirtualChannel.13.Main.UnitsScaler | Units scaler for totalisers | float32 | 2203 | 8707 | 1dp |
| VirtualChannel.13.Trend.Colour | As VirtualChannel1.Trend.Colour | uint8 | 2220 | 8736 | Not applicable |
| VirtualChannel.13.Trend.SpanHigh | Specifies the highest PV (output value) to be displayed | float32 | 2222 | 8738 | Same as VirtualChannel.13.Main.PV |
| VirtualChannel.13.Trend.SpanLow | Specifies the lowest PV (output value) to be displayed | float32 | 2221 | 8737 | Same as VirtualChannel.13.Main.PV |
| VirtualChannel.14.Alarm1.Acknowledge | 1 = acknowledge alarm | bool | 01da | 474 | Not applicable |
| VirtualChannel.14.Alarm1.Acknowledgement | 1 = alarm acknowledged | bool | 22d0 | 8912 | Not applicable |
| VirtualChannel.14.Alarm1.Active | 1 = alarm source active, or safe but not ack'd | bool | 22cb | 8907 | Not applicable |
| VirtualChannel.14.Alarm1.Amount | Rate-of-change alarm 'Amount' | float32 | 22c8 | 8904 | Same as VirtualChannel.14.Main.PV |
| VirtualChannel.14.Alarm1.AverageTime | Rate-of-change alarm 'Average time' | time_t | 22ca | 8906 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.14.Alarm1.Block | 0 = Blocking alarms off; 1 = Blocking alarms on | uint8 | 22c2 | 8898 | Not applicable |
| VirtualChannel.14.Alarm1.ChangeTime | Rate-of-change alarm 'Change Time' | uint8 | 22c9 | 8905 | Not applicable |
| VirtualChannel.14.Alarm1.Deviation | Deviation alarm 'Deviation Value' | float32 | 22c7 | 8903 | Same as VirtualChannel.14.Main.PV |
| VirtualChannel.14.Alarm1.Dwell | Alarm dwell time | time_t | 22c5 | 8901 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.14.Alarm1.Hysteresis | Alarm hysteresis value | float32 | 22c4 | 8900 | Same as VirtualChannel.14.Main.PV |
| VirtualChannel.14.Alarm1.Inactive | 1 = alarm source safe and ack'd (if necessary) | bool | 22ce | 8910 | Not applicable |
| VirtualChannel.14.Alarm1.Inhibit | 1 = alarm inhibited | bool | 22d1 | 8913 | Not applicable |
| VirtualChannel.14.Alarm1.Latch | As VirtualChannel1.Alarm1.Latch | uint8 | 22c1 | 8897 | Not applicable |
| VirtualChannel.14.Alarm1.NotAcknowledged | 1 = alarm has not been acknowledged | bool | 22cf | 8911 | Not applicable |
| VirtualChannel.14.Alarm1.Reference | Deviation alarm 'Reference' value | float32 | 22c6 | 8902 | Same as VirtualChannel.14.Main.PV |
| VirtualChannel.14.Alarm1.Status | As VirtualChannel1.Alarm1.Status | uint8 | 0156 | 342 | Not applicable |
| VirtualChannel.14.Alarm1.Threshold | Alarm trigger threshold | float32 | 22c3 | 8899 | Same as VirtualChannel.14.Main.PV |
| VirtualChannel.14.Alarm1.Type | As VirtualChannel1.Alarm1.Type | uint8 | 22c0 | 8896 | Not applicable |
| VirtualChannel.14.Alarm2.Acknowledge | 1 = acknowledge alarm | bool | 01db | 475 | Not applicable |
| VirtualChannel.14.Alarm2.Acknowledgement | 1 = alarm acknowledged | bool | 22f0 | 8944 | Not applicable |
| VirtualChannel.14.Alarm2.Active | 1 = alarm source active, or safe but not ack'd | bool | 22eb | 8939 | Not applicable |
| VirtualChannel.14.Alarm2.Amount | Rate-of-change alarm 'Amount' | float32 | 22e8 | 8936 | Same as VirtualChannel.14.Main.PV |
| VirtualChannel.14.Alarm2.AverageTime | Rate-of-change alarm 'Average time' | time_t | 22ea | 8938 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.14.Alarm2.Block | 0 = Blocking alarms off; 1 = Blocking alarms on | uint8 | 22e2 | 8930 | Not applicable |
| VirtualChannel.14.Alarm2.ChangeTime | Rate-of-change alarm 'Change Time' | uint8 | 22e9 | 8937 | Not applicable |
| VirtualChannel.14.Alarm2.Deviation | Deviation alarm 'Deviation Value' | float32 | 22e7 | 8935 | Same as VirtualChannel.14.Main.PV |
| VirtualChannel.14.Alarm2.Dwell | Alarm dwell time | time_t | 22e5 | 8933 | Set by Network.Modbus.TimeFormat |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| VirtualChannel.14.Alarm2.Hysteresis | Alarm hysteresis value | float32 | 22e4 | 8932 | Same as VirtualChannel.14.Main.PV |
| VirtualChannel.14.Alarm2.Inactive | 1 = alarm source safe and ack'd (if necessary) | bool | 22ee | 8942 | Not applicable |
| VirtualChannel.14.Alarm2.Inhibit | 1 = alarm inhibited | bool | 22f1 | 8945 | Not applicable |
| VirtualChannel.14.Alarm2.Latch | As VirtualChannel1.Alarm1.Latch | uint8 | 22e1 | 8929 | Not applicable |
| VirtualChannel.14.Alarm2.NotAcknowledged | 1 = alarm has not been acknowledged | bool | 22ef | 8943 | Not applicable |
| VirtualChannel.14.Alarm2.Reference | Deviation alarm 'Reference' value | float32 | 22e6 | 8934 | Same as VirtualChannel.14.Main.PV |
| VirtualChannel.14.Alarm2.Status | As VirtualChannel1.Alarm1.Status | uint8 | 0157 | 343 | Not applicable |
| VirtualChannel.14.Alarm2.Threshold | Alarm trigger threshold | float32 | 22e3 | 8931 | Same as VirtualChannel.14.Main.PV |
| VirtualChannel.14.Alarm2.Type | As VirtualChannel1.Alarm1.Type | uint8 | 22e0 | 8928 | Not applicable |
| VirtualChannel.14.Main.Descriptor | Virtual Channel descriptor | string_t | 4c5f | 19551 | Not applicable |
| VirtualChannel.14.Main.Disable | 1 = Virtual channel disabled | bool | 22a3 | 8867 | Not applicable |
| VirtualChannel.14.Main.HighCutOff | The highest input value that will be totalised/counted | float32 | 2285 | 8837 | Set by VirtualChannel.14.Main.Resolution |
| VirtualChannel.14.Main.Input1 | Input 1 value | float32 | 2287 | 8839 | Set by VirtualChannel.14.Main.Resolution |
| VirtualChannel.14.Main.Input2 | Input 2 value | float32 | 2288 | 8840 | Set by VirtualChannel.14.Main.Resolution |
| VirtualChannel.14.Main.LowCutOff | The lowest input value that will be totalised/counted | float32 | 2284 | 8836 | Set by VirtualChannel.14.Main.Resolution |
| VirtualChannel.14.Main.ModbusInput | Modbus input value | float32 | 2286 | 8838 | Set by VirtualChannel.14.Main.Resolution |
| VirtualChannel.14.Main.Operation | As VirtualChannel1.Main.Operation | uint8 | 2281 | 8833 | Not applicable |
| VirtualChannel.14.Main.Period | The time period over which the calculation is made | int32 | 228a | 8842 | Not applicable |
| VirtualChannel.14.Main.Preset | Initiate preset. 0 = No; 1 = Yes | bool | 228c | 8844 | Not applicable |
| VirtualChannel.14.Main.PresetValue | The preset value | float32 | 228d | 8845 | Set by VirtualChannel.14.Main.Resolution |
| VirtualChannel.14.Main.PV | The virtual channel output value | float32 | 0154 | 340 | Set by VirtualChannel.14.Main.Resolution |
| VirtualChannel.14.Main.Reset | Initiate reset. 0 = No; 1 = Yes | bool | 228b | 8843 | Not applicable |
| VirtualChannel.14.Main.Resolution | Number of decimal places (0 to 6) | uint8 | 2282 | 8834 | Not applicable |
| VirtualChannel.14.Main.Rollover | A pulse signal to indicate PV (output) has just rolled over | bool | 2291 | 8849 | Not applicable |
| VirtualChannel.14.Main.RolloverValue | Rollover value | float32 | 2292 | 8850 | Set by VirtualChannel.14.Main.Resolution |
| VirtualChannel.14.Main.Status | As VirtualChannel1.Main.Status | uint8 | 0155 | 341 | Not applicable |
| VirtualChannel.14.Main.TimeRemaining | Time remaining before the calculation is made | time_t | 2289 | 8841 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.14.Main.Trigger | Increment/decrement counter. 0 = No; 1 = Yes | bool | 228e | 8846 | Not applicable |
| VirtualChannel.14.Main.Type | As VirtualChannel1.Main.Type | uint8 | 2280 | 8832 | Not applicable |
| VirtualChannel.14.Main.Units | Units descriptor | string_t | 4c75 | 19573 | Not applicable |
| VirtualChannel.14.Main.UnitsScaler | Units scaler for totalisers | float32 | 2283 | 8835 | 1dp |
| VirtualChannel.14.Trend.Colour | As VirtualChannel1.Trend.Colour | uint8 | 22a0 | 8864 | Not applicable |
| VirtualChannel.14.Trend.SpanHigh | Specifies the highest PV (output value) to be displayed | float32 | 22a2 | 8866 | Same as VirtualChannel.14.Main.PV |
| VirtualChannel.14.Trend.SpanLow | Specifies the lowest PV (output value) to be displayed | float32 | 22a1 | 8865 | Same as VirtualChannel.14.Main.PV |
| VirtualChannel.15.Alarm1.Acknowledge | 1 = acknowledge alarm | bool | 01de | 478 | Not applicable |
| VirtualChannel.15.Alarm1.Acknowledgement | 1 = alarm acknowledged | bool | 2350 | 9040 | Not applicable |
| VirtualChannel.15.Alarm1.Active | 1 = alarm source active, or safe but not ack'd | bool | 234b | 9035 | Not applicable |
| VirtualChannel.15.Alarm1.Amount | Rate-of-change alarm 'Amount' | float32 | 2348 | 9032 | Same as VirtualChannel.15.Main.PV |
| VirtualChannel.15.Alarm1.AverageTime | Rate-of-change alarm 'Average time' | time_t | 234a | 9034 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.15.Alarm1.Block | 0 = Blocking alarms off; 1 = Blocking alarms on | uint8 | 2342 | 9026 | Not applicable |
| VirtualChannel.15.Alarm1.ChangeTime | Rate-of-change alarm 'Change Time' | uint8 | 2349 | 9033 | Not applicable |
| VirtualChannel.15.Alarm1.Deviation | Deviation alarm 'Deviation Value' | float32 | 2347 | 9031 | Same as VirtualChannel.15.Main.PV |
| VirtualChannel.15.Alarm1.Dwell | Alarm dwell time | time_t | 2345 | 9029 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.15.Alarm1.Hysteresis | Alarm hysteresis value | float32 | 2344 | 9028 | Same as VirtualChannel.15.Main.PV |
| VirtualChannel.15.Alarm1.Inactive | 1 = alarm source safe and ack'd (if necessary) | bool | 234e | 9038 | Not applicable |
| VirtualChannel.15.Alarm1.Inhibit | 1 = Alarm inhibited | bool | 2351 | 9041 | Not applicable |
| VirtualChannel.15.Alarm1.Latch | As VirtualChannel1.Alarm1.Latch | uint8 | 2341 | 9025 | Not applicable |
| VirtualChannel.15.Alarm1.NotAcknowledged | 1 = alarm has not been acknowledged | bool | 234f | 9039 | Not applicable |
| VirtualChannel.15.Alarm1.Reference | Deviation alarm 'Reference' value | float32 | 2346 | 9030 | Same as VirtualChannel.15.Main.PV |
| VirtualChannel.15.Alarm1.Status | As VirtualChannel1.Alarm1.Status | uint8 | 015a | 346 | Not applicable |
| VirtualChannel.15.Alarm1.Threshold | Alarm trigger threshold | float32 | 2343 | 9027 | Same as VirtualChannel.15.Main.PV |
| VirtualChannel.15.Alarm1.Type | As VirtualChannel1.Alarm1.Type | uint8 | 2340 | 9024 | Not applicable |
| VirtualChannel.15.Alarm2.Acknowledge | 1 = acknowledge alarm | bool | 01dd | 477 | Not applicable |
| VirtualChannel.15.Alarm2.Acknowledgement | 1 = alarm acknowledged | bool | 2370 | 9072 | Not applicable |
| VirtualChannel.15.Alarm2.Active | 1 = alarm source active, or safe but not ack'd | bool | 236b | 9067 | Not applicable |
| VirtualChannel.15.Alarm2.Amount | Rate-of-change alarm 'Amount' | float32 | 2368 | 9064 | Same as VirtualChannel.15.Main.PV |
| VirtualChannel.15.Alarm2.AverageTime | Rate-of-change alarm 'Average time' | time_t | 236a | 9066 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.15.Alarm2.Block | 0 = Blocking alarms off; 1 = Blocking alarms on | uint8 | 2362 | 9058 | Not applicable |
| VirtualChannel.15.Alarm2.ChangeTime | Rate-of-change alarm 'Change Time' | uint8 | 2369 | 9065 | Not applicable |
| VirtualChannel.15.Alarm2.Deviation | Deviation alarm 'Deviation Value' | float32 | 2367 | 9063 | Same as VirtualChannel.15.Main.PV |
| VirtualChannel.15.Alarm2.Dwell | Alarm dwell time | time_t | 2365 | 9061 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.15.Alarm2.Hysteresis | Alarm hysteresis value | float32 | 2364 | 9060 | Same as VirtualChannel.15.Main.PV |
| VirtualChannel.15.Alarm2.Inactive | 1 = alarm source safe and ack'd (if necessary) | bool | 236e | 9070 | Not applicable |
| VirtualChannel.15.Alarm2.Inhibit | 1 = alarm inhibited | bool | 2371 | 9073 | Not applicable |
| VirtualChannel.15.Alarm2.Latch | As VirtualChannel1.Alarm1.Latch | uint8 | 2361 | 9057 | Not applicable |
| VirtualChannel.15.Alarm2.NotAcknowledged | 1 = alarm has not been acknowledged | bool | 236f | 9071 | Not applicable |
| VirtualChannel.15.Alarm2.Reference | Deviation alarm 'Reference' value | float32 | 2366 | 9062 | Same as VirtualChannel.15.Main.PV |
| VirtualChannel.15.Alarm2.Status | As VirtualChannel1.Alarm1.Status | uint8 | 015b | 347 | Not applicable |
| VirtualChannel.15.Alarm2.Threshold | Alarm trigger threshold | float32 | 2363 | 9059 | Same as VirtualChannel.15.Main.PV |
| VirtualChannel.15.Alarm2.Type | As VirtualChannel1.Alarm1.Type | uint8 | 2360 | 9056 | Not applicable |
| VirtualChannel.15.Main.Descriptor | Virtual Channel descriptor | string_t | 4c7b | 19579 | Not applicable |
| VirtualChannel.15.Main.Disable | 1 = Virtual channel disabled | bool | 2323 | 8995 | Not applicable |
| VirtualChannel.15.Main.HighCutOff | The highest input value that will be totalised/counted | float32 | 2305 | 8965 | Set by VirtualChannel.15.Main.Resolution |
| VirtualChannel.15.Main.Input1 | Input 1 value | float32 | 2307 | 8967 | Set by VirtualChannel.15.Main.Resolution |
| VirtualChannel.15.Main.Input2 | Input 2 value | float32 | 2308 | 8968 | Set by VirtualChannel.15.Main.Resolution |
| VirtualChannel.15.Main.LowCutOff | The lowest input value that will be totalised/counted | float32 | 2304 | 8964 | Set by VirtualChannel.15.Main.Resolution |
| VirtualChannel.15.Main.ModbusInput | Modbus input value | float32 | 2306 | 8966 | Set by VirtualChannel.15.Main.Resolution |
| VirtualChannel.15.Main.Operation | Specifies the operation of the virtual channel | uint8 | 2301 | 8961 | Not applicable |
| VirtualChannel.15.Main.Period | The time period over which the calculation is made | int32 | 230a | 8970 | Not applicable |
| VirtualChannel.15.Main.Preset | Initiate preset. 0 = No; 1 = Yes | bool | 230c | 8972 | Not applicable |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| VirtualChannel.15.Main.PresetValue | Specifies the preset value | float32 | 230d | 8973 | Set by VirtualChannel.15.Main.Resolution |
| VirtualChannel.15.Main.PV | The virtual channel output value | float32 | 0158 | 344 | Set by VirtualChannel.15.Main.Resolution |
| VirtualChannel.15.Main.Reset | Initiate reset. 0 = No; 1 = Yes | bool | 230b | 8971 | Not applicable |
| VirtualChannel.15.Main.Resolution | Specifies the resolution/number of decimal places | uint8 | 2302 | 8962 | Not applicable |
| VirtualChannel.15.Main.Rollover | A pulse signal to indicate PV (output) has just rolled over | bool | 2311 | 8977 | Not applicable |
| VirtualChannel.15.Main.RolloverValue | Rollover value | float32 | 2312 | 8978 | Set by VirtualChannel.15.Main.Resolution |
| VirtualChannel.15.Main.Status | As VirtualChannel1.Main.Status | uint8 | 0159 | 345 | Not applicable |
| VirtualChannel.15.Main.TimeRemaining | Time remaining before the calculation is made | time_t | 2309 | 8969 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.15.Main.Trigger | Increment/decrement counter. 0 = No; 1 = Yes | bool | 230e | 8974 | Not applicable |
| VirtualChannel.15.Main.Type | As VirtualChannel1.Main.Type | uint8 | 2300 | 8960 | Not applicable |
| VirtualChannel.15.Main.Units | Units descriptor | string_t | 4c90 | 19600 | Not applicable |
| VirtualChannel.15.Main.UnitsScaler | Units scaler for totalisers | float32 | 2303 | 8963 | 1dp |
| VirtualChannel.15.Trend.Colour | As VirtualChannel1.Trend.Colour | uint8 | 2320 | 8992 | Not applicable |
| VirtualChannel.15.Trend.SpanHigh | Specifies the highest PV (output value) to be displayed | float32 | 2322 | 8994 | Same as VirtualChannel.15.Main.PV |
| VirtualChannel.15.Trend.SpanLow | Specifies the lowest PV (output value) to be displayed | float32 | 2321 | 8993 | Same as VirtualChannel.15.Main.PV |
| VirtualChannel.16.Main.Descriptor | Virtual Channel descriptor | string_t | 4c96 | 19606 | Not applicable |
| VirtualChannel.16.Main.Disable | 1 = Virtual channel disabled | bool | 23a3 | 9123 | Not applicable |
| VirtualChannel.16.Main.HighCutOff | The highest input value that will be totalised/counted | float32 | 2385 | 9093 | Set by VirtualChannel.16.Main.Resolution |
| VirtualChannel.16.Main.Input1 | Input 1 value | float32 | 2387 | 9095 | Set by VirtualChannel.16.Main.Resolution |
| VirtualChannel.16.Main.Input2 | Input 2 value | float32 | 2388 | 9096 | Set by VirtualChannel.16.Main.Resolution |
| VirtualChannel.16.Main.LowCutOff | The lowest input value that will be totalised/counted | float32 | 2384 | 9092 | Set by VirtualChannel.16.Main.Resolution |
| VirtualChannel.16.Main.ModbusInput | Modbus input value | float32 | 2386 | 9094 | Set by VirtualChannel.16.Main.Resolution |
| VirtualChannel.16.Main.Operation | Specifies the operation of the virtual channel | uint8 | 2381 | 9089 | Not applicable |
| VirtualChannel.16.Main.Period | The time period over which the calculation is made | int32 | 238a | 9098 | Not applicable |
| VirtualChannel.16.Main.Preset | Initiate preset. 0 = No; 1 = Yes | bool | 238c | 9100 | Not applicable |
| VirtualChannel.16.Main.PresetValue | Specifies the preset value | float32 | 238d | 9101 | Set by VirtualChannel.16.Main.Resolution |
| VirtualChannel.16.Main.PV | The virtual channel output value | float32 | 015c | 348 | Set by VirtualChannel.16.Main.Resolution |
| VirtualChannel.16.Main.Reset | Initiate reset. 0 = No; 1 = Yes | bool | 238b | 9099 | Not applicable |
| VirtualChannel.16.Main.Resolution | Specifies the resolution/number of decimal places | uint8 | 2382 | 9090 | Not applicable |
| VirtualChannel.16.Main.Rollover | A pulse signal to indicate PV (output) has just rolled over | bool | 2391 | 9105 | Not applicable |
| VirtualChannel.16.Main.RolloverValue | Rollover value | float32 | 2392 | 9106 | Set by VirtualChannel.16.Main.Resolution |
| VirtualChannel.16.Main.Status | As VirtualChannel1.Main.Status | uint8 | 015d | 349 | Not applicable |
| VirtualChannel.16.Main.TimeRemaining | Time remaining before the calculation is made | time_t | 2389 | 9097 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.16.Main.Trigger | Increment/decrement counter. 0 = No; 1 = Yes | bool | 238e | 9102 | Not applicable |
| VirtualChannel.16.Main.Type | As VirtualChannel1.Main.Type | uint8 | 2380 | 9088 | Not applicable |
| VirtualChannel.16.Main.Units | Units descriptor | string_t | 4cab | 19627 | Not applicable |
| VirtualChannel.16.Main.UnitsScaler | Units scaler for totalisers | float32 | 2383 | 9091 | 1dp |
| VirtualChannel.16.Trend.Colour | As VirtualChannel1.Trend.Colour | uint8 | 23a0 | 9120 | Not applicable |
| VirtualChannel.16.Trend.SpanHigh | Specifies the highest PV (output value) to be displayed | float32 | 23a2 | 9122 | Same as VirtualChannel.16.Main.PV |
| VirtualChannel.16.Trend.SpanLow | Specifies the lowest PV (output value) to be displayed | float32 | 23a1 | 9121 | Same as VirtualChannel.16.Main.PV |
| VirtualChannel.17.Main.Descriptor | Virtual Channel descriptor | string_t | 4cb1 | 19633 | Not applicable |
| VirtualChannel.17.Main.Disable | 1 = Virtual channel disabled | bool | 23e3 | 9187 | Not applicable |
| VirtualChannel.17.Main.HighCutOff | The highest input value that will be totalised/counted | float32 | 23c5 | 9157 | Set by VirtualChannel.17.Main.Resolution |
| VirtualChannel.17.Main.Input1 | Input 1 value | float32 | 23c7 | 9159 | Set by VirtualChannel.17.Main.Resolution |
| VirtualChannel.17.Main.Input2 | Input 2 value | float32 | 23c8 | 9160 | Set by VirtualChannel.17.Main.Resolution |
| VirtualChannel.17.Main.LowCutOff | The lowest input value that will be totalised/counted | float32 | 23c4 | 9156 | Set by VirtualChannel.17.Main.Resolution |
| VirtualChannel.17.Main.ModbusInput | Modbus input value | float32 | 23c6 | 9158 | Set by VirtualChannel.17.Main.Resolution |
| VirtualChannel.17.Main.Operation | Specifies the operation of the virtual channel | uint8 | 23c1 | 9153 | Not applicable |
| VirtualChannel.17.Main.Period | The time period over which the calculation is made | int32 | 23ca | 9162 | Not applicable |
| VirtualChannel.17.Main.Preset | Initiate preset. 0 = No; 1 = Yes | bool | 23cc | 9164 | Not applicable |
| VirtualChannel.17.Main.PresetValue | Specifies the preset value | float32 | 23cd | 9165 | Set by VirtualChannel.17.Main.Resolution |
| VirtualChannel.17.Main.PV | The virtual channel output value | float32 | 015e | 350 | Set by VirtualChannel.17.Main.Resolution |
| VirtualChannel.17.Main.Reset | Initiate reset. 0 = No; 1 = Yes | bool | 23cb | 9163 | Not applicable |
| VirtualChannel.17.Main.Resolution | Specifies the resolution/number of decimal places | uint8 | 23c2 | 9154 | Not applicable |
| VirtualChannel.17.Main.Rollover | A pulse signal to indicate PV (output) has just rolled over | bool | 23d1 | 9169 | Not applicable |
| VirtualChannel.17.Main.RolloverValue | Rollover value | float32 | 23d2 | 9170 | Set by VirtualChannel.17.Main.Resolution |
| VirtualChannel.17.Main.Status | As VirtualChannel1.Main.Status | uint8 | 015f | 351 | Not applicable |
| VirtualChannel.17.Main.TimeRemaining | Time remaining before the calculation is made | time_t | 23c9 | 9161 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.17.Main.Trigger | Increment/decrement counter. 0 = No; 1 = Yes | bool | 23ce | 9166 | Not applicable |
| VirtualChannel.17.Main.Type | As VirtualChannel1.Main.Type | uint8 | 23c0 | 9152 | Not applicable |
| VirtualChannel.17.Main.Units | Units descriptor | string_t | 4cc6 | 19654 | Not applicable |
| VirtualChannel.17.Main.UnitsScaler | Units scaler for totalisers | float32 | 23c3 | 9155 | 1dp |
| VirtualChannel.17.Trend.Colour | As VirtualChannel1.Trend.Colour | uint8 | 23e0 | 9184 | Not applicable |
| VirtualChannel.17.Trend.SpanHigh | Specifies the highest PV (output value) to be displayed | float32 | 23e2 | 9186 | Same as VirtualChannel.17.Main.PV |
| VirtualChannel.17.Trend.SpanLow | Specifies the lowest PV (output value) to be displayed | float32 | 23e1 | 9185 | Same as VirtualChannel.17.Main.PV |
| VirtualChannel.18.Main.Descriptor | Virtual Channel descriptor | string_t | 4ccc | 19660 | Not applicable |
| VirtualChannel.18.Main.Disable | 1 = Virtual channel disabled | bool | 2523 | 9507 | Not applicable |
| VirtualChannel.18.Main.HighCutOff | The highest input value that will be totalised/counted | float32 | 2405 | 9221 | Set by VirtualChannel.18.Main.Resolution |
| VirtualChannel.18.Main.Input1 | Input 1 value | float32 | 2407 | 9223 | Set by VirtualChannel.18.Main.Resolution |
| VirtualChannel.18.Main.Input2 | Input 2 value | float32 | 2408 | 9224 | Set by VirtualChannel.18.Main.Resolution |
| VirtualChannel.18.Main.LowCutOff | The lowest input value that will be totalised/counted | float32 | 2404 | 9220 | Set by VirtualChannel.18.Main.Resolution |
| VirtualChannel.18.Main.ModbusInput | Modbus input value | float32 | 2406 | 9222 | Set by VirtualChannel.18.Main.Resolution |
| VirtualChannel.18.Main.Operation | Specifies the operation of the virtual channel | uint8 | 2401 | 9217 | Not applicable |
| VirtualChannel.18.Main.Period | The time period over which the calculation is made | int32 | 240a | 9226 | Not applicable |
| VirtualChannel.18.Main.Preset | Initiate preset. 0 = No; 1 = Yes | bool | 240c | 9228 | Not applicable |
| VirtualChannel.18.Main.PresetValue | Specifies the preset value | float32 | 240d | 9229 | Set by VirtualChannel.18.Main.Resolution |
| VirtualChannel.18.Main.PV | The virtual channel output value | float32 | 0160 | 352 | Set by VirtualChannel.18.Main.Resolution |
| VirtualChannel.18.Main.Reset | Initiate reset. 0 = No; 1 = Yes | bool | 240b | 9227 | Not applicable |
| VirtualChannel.18.Main.Resolution | Specifies the resolution/number of decimal places | uint8 | 2402 | 9218 | Not applicable |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| VirtualChannel.18.Main.Rollover | A pulse signal to indicate PV (output) has just rolled over | bool | 2411 | 9233 | Not applicable |
| VirtualChannel.18.Main.RolloverValue | Rollover value | float32 | 2412 | 9234 | Set by VirtualChannel.18.Main.Resolution |
| VirtualChannel.18.Main.Status | As VirtualChannel1.Main.Status | uint8 | 0161 | 353 | Not applicable |
| VirtualChannel.18.Main.TimeRemaining | Time remaining before the calculation is made | time_t | 2409 | 9225 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.18.Main.Trigger | Increment/decrement counter. 0 = No; 1 = Yes) | bool | 240e | 9230 | Not applicable |
| VirtualChannel.18.Main.Type | As VirtualChannel1.Main.Type | uint8 | 2400 | 9216 | Not applicable |
| VirtualChannel.18.Main.Units | Units descriptor | string_t | 4ce1 | 19681 | Not applicable |
| VirtualChannel.18.Main.UnitsScaler | Units scaler for totalisers | float32 | 2403 | 9219 | 1dp |
| VirtualChannel.18.Trend.Colour | As VirtualChannel1.Trend.Colour | uint8 | 2520 | 9504 | Not applicable |
| VirtualChannel.18.Trend.SpanHigh | Specifies the highest PV (output value) to be displayed | float32 | 2522 | 9506 | Same as VirtualChannel.18.Main.PV |
| VirtualChannel.18.Trend.SpanLow | Specifies the lowest PV (output value) to be displayed | float32 | 2521 | 9505 | Same as VirtualChannel.18.Main.PV |
| VirtualChannel.19.Main.Descriptor | Virtual Channel descriptor | string_t | 4ce7 | 19687 | Not applicable |
| VirtualChannel.19.Main.Disable | 1 = Virtual channel disabled | bool | 2563 | 9571 | Not applicable |
| VirtualChannel.19.Main.HighCutOff | The highest input value that will be totalised/counted | float32 | 2445 | 9285 | Set by VirtualChannel.19.Main.Resolution |
| VirtualChannel.19.Main.Input1 | Input 1 value | float32 | 2447 | 9287 | Set by VirtualChannel.19.Main.Resolution |
| VirtualChannel.19.Main.Input2 | Input 2 value | float32 | 2448 | 9288 | Set by VirtualChannel.19.Main.Resolution |
| VirtualChannel.19.Main.LowCutOff | The lowest input value that will be totalised/counted | float32 | 2444 | 9284 | Set by VirtualChannel.19.Main.Resolution |
| VirtualChannel.19.Main.ModbusInput | Modbus input value | float32 | 2446 | 9286 | Set by VirtualChannel.19.Main.Resolution |
| VirtualChannel.19.Main.Operation | Specifies the operation of the virtual channel | uint8 | 2441 | 9281 | Not applicable |
| VirtualChannel.19.Main.Period | The time period over which the calculation is made | int32 | 244a | 9290 | Not applicable |
| VirtualChannel.19.Main.Preset | Initiate preset. 0 = No; 1 = Yes | bool | 244c | 9292 | Not applicable |
| VirtualChannel.19.Main.PresetValue | Specifies the preset value | float32 | 244d | 9293 | Set by VirtualChannel.19.Main.Resolution |
| VirtualChannel.19.Main.PV | The virtual channel output value | float32 | 0162 | 354 | Set by VirtualChannel.19.Main.Resolution |
| VirtualChannel.19.Main.Reset | Initiate reset. 0 = No; 1 = Yes | bool | 244b | 9291 | Not applicable |
| VirtualChannel.19.Main.Resolution | Specifies the resolution/number of decimal places | uint8 | 2442 | 9282 | Not applicable |
| VirtualChannel.19.Main.Rollover | A pulse signal to indicate PV (output) has just rolled over | bool | 2451 | 9297 | Not applicable |
| VirtualChannel.19.Main.RolloverValue | Rollover value | float32 | 2452 | 9298 | Set by VirtualChannel.19.Main.Resolution |
| VirtualChannel.19.Main.Status | TAs VirtualChannel1.Main.Statusv | uint8 | 0163 | 355 | Not applicable |
| VirtualChannel.19.Main.TimeRemaining | Time remaining before the calculation is made | time_t | 2449 | 9289 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.19.Main.Trigger | Increment/decrement counter. 0 = No; 1 = Yes | bool | 244e | 9294 | Not applicable |
| VirtualChannel.19.Main.Type | As VirtualChannel1.Main.Type | uint8 | 2440 | 9280 | Not applicable |
| VirtualChannel.19.Main.Units | Units descriptor | string_t | 4cfc | 19708 | Not applicable |
| VirtualChannel.19.Main.UnitsScaler | Units scaler for totalisers | float32 | 2443 | 9283 | 1dp |
| VirtualChannel.19.Trend.Colour | As VirtualChannel1.Trend.Colour | uint8 | 2560 | 9568 | Not applicable |
| VirtualChannel.19.Trend.SpanHigh | Specifies the highest PV (output value) to be displayed | float32 | 2562 | 9570 | Same as VirtualChannel.19.Main.PV |
| VirtualChannel.19.Trend.SpanLow | Specifies the lowest PV (output value) to be displayed | float32 | 2561 | 9569 | Same as VirtualChannel.19.Main.PV |
| VirtualChannel.20.Main.Descriptor | Virtual Channel descriptor | string_t | 4d02 | 19714 | Not applicable |
| VirtualChannel.20.Main.Disable | 1 = Virtual channel disabled | bool | 25a3 | 9635 | Not applicable |
| VirtualChannel.20.Main.HighCutOff | The highest input value that will be totalised/counted | float32 | 2485 | 9349 | Set by VirtualChannel.20.Main.Resolution |
| VirtualChannel.20.Main.Input1 | Input 1 value | float32 | 2487 | 9351 | Set by VirtualChannel.20.Main.Resolution |
| VirtualChannel.20.Main.Input2 | Input 2 value | float32 | 2488 | 9352 | Set by VirtualChannel.20.Main.Resolution |
| VirtualChannel.20.Main.LowCutOff | The lowest input value that will be totalised/counted | float32 | 2484 | 9348 | Set by VirtualChannel.20.Main.Resolution |
| VirtualChannel.20.Main.ModbusInput | Modbus input value | float32 | 2486 | 9350 | Set by VirtualChannel.20.Main.Resolution |
| VirtualChannel.20.Main.Operation | Specifies the operation of the virtual channel | uint8 | 2481 | 9345 | Not applicable |
| VirtualChannel.20.Main.Period | The time period over which the calculation is made | int32 | 248a | 9354 | Not applicable |
| VirtualChannel.20.Main.Preset | Initiate preset. 0 = No; 1 = Yes | bool | 248c | 9356 | Not applicable |
| VirtualChannel.20.Main.PresetValue | Specifies the preset value | float32 | 248d | 9357 | Set by VirtualChannel.20.Main.Resolution |
| VirtualChannel.20.Main.PV | The virtual channel output value | float32 | 0164 | 356 | Set by VirtualChannel.20.Main.Resolution |
| VirtualChannel.20.Main.Reset | Initiate reset. 0 = No; 1 = Yes | bool | 248b | 9355 | Not applicable |
| VirtualChannel.20.Main.Resolution | Specifies the resolution/number of decimal places | uint8 | 2482 | 9346 | Not applicable |
| VirtualChannel.20.Main.Rollover | A pulse signal to indicate PV (output) has just rolled over | bool | 2491 | 9361 | Not applicable |
| VirtualChannel.20.Main.RolloverValue | Rollover value | float32 | 2492 | 9362 | Set by VirtualChannel.20.Main.Resolution |
| VirtualChannel.20.Main.Status | As VirtualChannel1.Main.Status | uint8 | 0165 | 357 | Not applicable |
| VirtualChannel.20.Main.TimeRemaining | Time remaining before the calculation is made | time_t | 2489 | 9353 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.20.Main.Trigger | TIncrement/decrement counter. 0 = No; 1 = Yes | bool | 248e | 9358 | Not applicable |
| VirtualChannel.20.Main.Type | As VirtualChannel1.Main.Type | uint8 | 2480 | 9344 | Not applicable |
| VirtualChannel.20.Main.Units | Units descriptor | string_t | 4d17 | 19735 | Not applicable |
| VirtualChannel.20.Main.UnitsScaler | Units scaler for totalisers | float32 | 2483 | 9347 | 1dp |
| VirtualChannel.20.Trend.Colour | As VirtualChannel1.Trend.Colour | uint8 | 25a0 | 9632 | Not applicable |
| VirtualChannel.20.Trend.SpanHigh | Specifies the highest PV (output value) to be displayed | float32 | 25a2 | 9634 | Same as VirtualChannel.20.Main.PV |
| VirtualChannel.20.Trend.SpanLow | Specifies the lowest PV (output value) to be displayed | float32 | 25a1 | 9633 | Same as VirtualChannel.20.Main.PV |
| VirtualChannel.21.Main.Descriptor | Virtual Channel descriptor | string_t | 4d1d | 19741 | Not applicable |
| VirtualChannel.21.Main.Disable | 1 = Virtual channel disabled | bool | 25e3 | 9699 | Not applicable |
| VirtualChannel.21.Main.HighCutOff | The highest input value that will be totalised/counted | float32 | 24c5 | 9413 | Set by VirtualChannel.21.Main.Resolution |
| VirtualChannel.21.Main.Input1 | Input 1 value | float32 | 24c7 | 9415 | Set by VirtualChannel.21.Main.Resolution |
| VirtualChannel.21.Main.Input2 | Input 2 value | float32 | 24c8 | 9416 | Set by VirtualChannel.21.Main.Resolution |
| VirtualChannel.21.Main.LowCutOff | The lowest input value that will be totalised/counted | float32 | 24c4 | 9412 | Set by VirtualChannel.21.Main.Resolution |
| VirtualChannel.21.Main.ModbusInput | Modbus input value | float32 | 24c6 | 9414 | Set by VirtualChannel.21.Main.Resolution |
| VirtualChannel.21.Main.Operation | Specifies the operation of the virtual channel | uint8 | 24c1 | 9409 | Not applicable |
| VirtualChannel.21.Main.Period | The time period over which the calculation is made | int32 | 24ca | 9418 | Not applicable |
| VirtualChannel.21.Main.Preset | Initiate preset. 0 = No; 1 = Yes | bool | 24cc | 9420 | Not applicable |
| VirtualChannel.21.Main.PresetValue | Specifies the preset value | float32 | 24cd | 9421 | Set by VirtualChannel.21.Main.Resolution |
| VirtualChannel.21.Main.PV | TThe virtual channel output value | float32 | 0166 | 358 | Set by VirtualChannel.21.Main.Resolution |
| VirtualChannel.21.Main.Reset | Initiate reset. 0 = No; 1 = Yes | bool | 24cb | 9419 | Not applicable |
| VirtualChannel.21.Main.Resolution | Specifies the resolution/number of decimal places | uint8 | 24c2 | 9410 | Not applicable |
| VirtualChannel.21.Main.Rollover | A pulse signal to indicate PV (output) has just rolled over | bool | 24d1 | 9425 | Not applicable |
| VirtualChannel.21.Main.RolloverValue | Rollover value | float32 | 24d2 | 9426 | Set by VirtualChannel.21.Main.Resolution |
| VirtualChannel.21.Main.Status | As VirtualChannel1.Main.Status | uint8 | 0167 | 359 | Not applicable |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| VirtualChannel.21.Main.TimeRemaining | Time remaining before the calculation is made | time_t | 24c9 | 9417 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.21.Main.Trigger | Increment/decrement counter. 0 = No; 1 = Yes | bool | 24ce | 9422 | Not applicable |
| VirtualChannel.21.Main.Type | As VirtualChannel1.Main.Type | uint8 | 24c0 | 9408 | Not applicable |
| VirtualChannel.21.Main.Units | Units descriptor | string_t | 4d32 | 19762 | Not applicable |
| VirtualChannel.21.Main.UnitsScaler | Units scaler for totalisers | float32 | 24c3 | 9411 | 1dp |
| VirtualChannel.21.Trend.Colour | As VirtualChannel1.Trend.Colour | uint8 | 25e0 | 9696 | Not applicable |
| VirtualChannel.21.Trend.SpanHigh | Specifies the highest PV (output value) to be displayed | float32 | 25e2 | 9698 | Same as VirtualChannel.21.Main.PV |
| VirtualChannel.21.Trend.SpanLow | Specifies the lowest PV (output value) to be displayed | float32 | 25e1 | 9697 | Same as VirtualChannel.21.Main.PV |
| | | | | | |
| VirtualChannel.22.Main.Descriptor | Virtual Channel descriptor | string_t | 4d38 | 19768 | Not applicable |
| VirtualChannel.22.Main.Disable | 1 = Virtual channel disabled | bool | 2623 | 9763 | Not applicable |
| VirtualChannel.22.Main.HighCutOff | The highest input value that will be totalised/counted | float32 | 2505 | 9477 | Set by VirtualChannel.22.Main.Resolution |
| VirtualChannel.22.Main.Input1 | Input 1 value | float32 | 2507 | 9479 | Set by VirtualChannel.22.Main.Resolution |
| VirtualChannel.22.Main.Input2 | Input 2 value | float32 | 2508 | 9480 | Set by VirtualChannel.22.Main.Resolution |
| VirtualChannel.22.Main.LowCutOff | The lowest input value that will be totalised/counted | float32 | 2504 | 9476 | Set by VirtualChannel.22.Main.Resolution |
| VirtualChannel.22.Main.ModbusInput | Modbus input value | float32 | 2506 | 9478 | Set by VirtualChannel.22.Main.Resolution |
| VirtualChannel.22.Main.Operation | Specifies the operation of the virtual channel | uint8 | 2501 | 9473 | Not applicable |
| VirtualChannel.22.Main.Period | The time period over which the calculation is made | int32 | 250a | 9482 | Not applicable |
| VirtualChannel.22.Main.Preset | Initiate preset. 0 = No; 1 = Yes | bool | 250c | 9484 | Not applicable |
| VirtualChannel.22.Main.PresetValue | Specifies the preset value | float32 | 250d | 9485 | Set by VirtualChannel.22.Main.Resolution |
| VirtualChannel.22.Main.PV | TThe virtual channel output value | float32 | 0168 | 360 | Set by VirtualChannel.22.Main.Resolution |
| VirtualChannel.22.Main.Reset | Initiate reset. 0 = No; 1 = Yes | bool | 250b | 9483 | Not applicable |
| VirtualChannel.22.Main.Resolution | Specifies the resolution/number of decimal places | uint8 | 2502 | 9474 | Not applicable |
| VirtualChannel.22.Main.Rollover | A pulse signal to indicate PV (output) has just rolled over | bool | 2511 | 9489 | Not applicable |
| VirtualChannel.22.Main.RolloverValue | Rollover value | float32 | 2512 | 9490 | Set by VirtualChannel.22.Main.Resolution |
| VirtualChannel.22.Main.Status | As VirtualChannel1.Main.Status | uint8 | 0169 | 361 | Not applicable |
| VirtualChannel.22.Main.TimeRemaining | Time remaining before the calculation is made | time_t | 2509 | 9481 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.22.Main.Trigger | Increment/decrement counter. 0 = No; 1 = Yes | bool | 250e | 9486 | Not applicable |
| VirtualChannel.22.Main.Type | As VirtualChannel1.Main.Type | uint8 | 2500 | 9472 | Not applicable |
| VirtualChannel.22.Main.Units | Units descriptor | string_t | 4d4d | 19789 | Not applicable |
| VirtualChannel.22.Main.UnitsScaler | Units scaler for totalisers | float32 | 2503 | 9475 | 1dp |
| VirtualChannel.22.Trend.Colour | As VirtualChannel1.Trend.Colour | uint8 | 2620 | 9760 | Not applicable |
| VirtualChannel.22.Trend.SpanHigh | Specifies the highest PV (output value) to be displayed | float32 | 2622 | 9762 | Same as VirtualChannel.22.Main.PV |
| VirtualChannel.22.Trend.SpanLow | Specifies the lowest PV (output value) to be displayed | float32 | 2621 | 9761 | Same as VirtualChannel.22.Main.PV |
| | | | | | |
| VirtualChannel.23.Main.Descriptor | Virtual Channel descriptor | string_t | 4d53 | 19795 | Not applicable |
| VirtualChannel.23.Main.Disable | 1 = Virtual channel disabled | bool | 2663 | 9827 | Not applicable |
| VirtualChannel.23.Main.HighCutOff | The highest input value that will be totalised/counted | float32 | 2545 | 9541 | Set by VirtualChannel.23.Main.Resolution |
| VirtualChannel.23.Main.Input1 | Input 1 value | float32 | 2547 | 9543 | Set by VirtualChannel.23.Main.Resolution |
| VirtualChannel.23.Main.Input2 | Input 2 value | float32 | 2548 | 9544 | Set by VirtualChannel.23.Main.Resolution |
| VirtualChannel.23.Main.LowCutOff | The lowest input value that will be totalised/counted | float32 | 2544 | 9540 | Set by VirtualChannel.23.Main.Resolution |
| VirtualChannel.23.Main.ModbusInput | Modbus input value | float32 | 2546 | 9542 | Set by VirtualChannel.23.Main.Resolution |
| VirtualChannel.23.Main.Operation | Specifies the operation of the virtual channel | uint8 | 2541 | 9537 | Not applicable |
| VirtualChannel.23.Main.Period | The time period over which the calculation is made | int32 | 254a | 9546 | Not applicable |
| VirtualChannel.23.Main.Preset | Initiate preset. 0 = No; 1 = Yes | bool | 254c | 9548 | Not applicable |
| VirtualChannel.23.Main.PresetValue | Specifies the preset value | float32 | 254d | 9549 | Set by VirtualChannel.23.Main.Resolution |
| VirtualChannel.23.Main.PV | The virtual channel output value | float32 | 016a | 362 | Set by VirtualChannel.23.Main.Resolution |
| VirtualChannel.23.Main.Reset | Initiate reset. 0 = No; 1 = Yes | bool | 254b | 9547 | Not applicable |
| VirtualChannel.23.Main.Resolution | Specifies the resolution/number of decimal places | uint8 | 2542 | 9538 | Not applicable |
| VirtualChannel.23.Main.Rollover | A pulse signal to indicate PV (output) has just rolled over | bool | 2551 | 9553 | Not applicable |
| VirtualChannel.23.Main.RolloverValue | Rollover value | float32 | 2552 | 9554 | Set by VirtualChannel.23.Main.Resolution |
| VirtualChannel.23.Main.Status | As VirtualChannel1.Main.Status | uint8 | 016b | 363 | Not applicable |
| VirtualChannel.23.Main.TimeRemaining | Time remaining before the calculation is made | time_t | 2549 | 9545 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.23.Main.Trigger | Increment/decrement counter. 0 = No; 1 = Yes | bool | 254e | 9550 | Not applicable |
| VirtualChannel.23.Main.Type | As VirtualChannel1.Main.Type | uint8 | 2540 | 9536 | Not applicable |
| VirtualChannel.23.Main.Units | Units descriptor | string_t | 4d68 | 19816 | Not applicable |
| VirtualChannel.23.Main.UnitsScaler | Units scaler for totalisers | float32 | 2543 | 9539 | 1dp |
| VirtualChannel.23.Trend.Colour | As VirtualChannel1.Trend.Colour | uint8 | 2660 | 9824 | Not applicable |
| VirtualChannel.23.Trend.SpanHigh | Specifies the highest PV (output value) to be displayed | float32 | 2662 | 9826 | Same as VirtualChannel.23.Main.PV |
| VirtualChannel.23.Trend.SpanLow | Specifies the lowest PV (output value) to be displayed | float32 | 2661 | 9825 | Same as VirtualChannel.23.Main.PV |
| | | | | | |
| VirtualChannel.24.Main.Descriptor | Virtual Channel descriptor | string_t | 4d6e | 19822 | Not applicable |
| VirtualChannel.24.Main.Disable | 1 = Virtual channel disabled | bool | 26a3 | 9891 | Not applicable |
| VirtualChannel.24.Main.HighCutOff | The highest input value that will be totalised/counted | float32 | 2585 | 9605 | Set by VirtualChannel.24.Main.Resolution |
| VirtualChannel.24.Main.Input1 | Input 1 value | float32 | 2587 | 9607 | Set by VirtualChannel.24.Main.Resolution |
| VirtualChannel.24.Main.Input2 | Input 2 value | float32 | 2588 | 9608 | Set by VirtualChannel.24.Main.Resolution |
| VirtualChannel.24.Main.LowCutOff | The lowest input value that will be totalised/counted | float32 | 2584 | 9604 | Set by VirtualChannel.24.Main.Resolution |
| VirtualChannel.24.Main.ModbusInput | Modbus input value | float32 | 2586 | 9606 | Set by VirtualChannel.24.Main.Resolution |
| VirtualChannel.24.Main.Operation | Specifies the operation of the virtual channel | uint8 | 2581 | 9601 | Not applicable |
| VirtualChannel.24.Main.Period | The time period over which the calculation is made | int32 | 258a | 9610 | Not applicable |
| VirtualChannel.24.Main.Preset | Initiate preset. 0 = No; 1 = Yes | bool | 258c | 9612 | Not applicable |
| VirtualChannel.24.Main.PresetValue | Specifies the preset value | float32 | 258d | 9613 | Set by VirtualChannel.24.Main.Resolution |
| VirtualChannel.24.Main.PV | The virtual channel output value | float32 | 016c | 364 | Set by VirtualChannel.24.Main.Resolution |
| VirtualChannel.24.Main.Reset | Initiate reset. 0 = No; 1 = Yes | bool | 258b | 9611 | Not applicable |
| VirtualChannel.24.Main.Resolution | Specifies the resolution/number of decimal places | uint8 | 2582 | 9602 | Not applicable |
| VirtualChannel.24.Main.Rollover | A pulse signal to indicate PV (output) has just rolled over | bool | 2591 | 9617 | Not applicable |
| VirtualChannel.24.Main.RolloverValue | Rollover value | float32 | 2592 | 9618 | Set by VirtualChannel.24.Main.Resolution |
| VirtualChannel.24.Main.Status | As VirtualChannel1.Main.Status | uint8 | 016d | 365 | Not applicable |
| VirtualChannel.24.Main.TimeRemaining | Time remaining before the calculation is made | time_t | 2589 | 9609 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.24.Main.Trigger | Increment/decrement counter. 0 = No; 1 = Yes | bool | 258e | 9614 | Not applicable |
| VirtualChannel.24.Main.Type | As VirtualChannel1.Main.Type | uint8 | 2580 | 9600 | Not applicable |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| VirtualChannel.24.Main.Units | Units descriptor | string_t | 4d83 | 19843 | Not applicable |
| VirtualChannel.24.Main.UnitsScaler | Units scaler for totalisers | float32 | 2583 | 9603 | 1dp |
| VirtualChannel.24.Trend.Colour | As VirtualChannel1.Trend.Colour | uint8 | 26a0 | 9888 | Not applicable |
| VirtualChannel.24.Trend.SpanHigh | Specifies the highest PV (output value) to be displayed | float32 | 26a2 | 9890 | Same as VirtualChannel.24.Main.PV |
| VirtualChannel.24.Trend.SpanLow | Specifies the lowest PV (output value) to be displayed | float32 | 26a1 | 9889 | Same as VirtualChannel.24.Main.PV |
| VirtualChannel.25.Main.Descriptor | Virtual Channel descriptor | string_t | 4d89 | 19849 | Not applicable |
| VirtualChannel.25.Main.Disable | 1 = Virtual channel disabled | bool | 26e3 | 9955 | Not applicable |
| VirtualChannel.25.Main.HighCutOff | The highest input value that will be totalised/counted | float32 | 25c5 | 9669 | Set by VirtualChannel.25.Main.Resolution |
| VirtualChannel.25.Main.Input1 | Input 1 value | float32 | 25c7 | 9671 | Set by VirtualChannel.25.Main.Resolution |
| VirtualChannel.25.Main.Input2 | Input 2 value | float32 | 25c8 | 9672 | Set by VirtualChannel.25.Main.Resolution |
| VirtualChannel.25.Main.LowCutOff | The lowest input value that will be totalised/counted | float32 | 25c4 | 9668 | Set by VirtualChannel.25.Main.Resolution |
| VirtualChannel.25.Main.ModbusInput | Modbus input value | float32 | 25c6 | 9670 | Set by VirtualChannel.25.Main.Resolution |
| VirtualChannel.25.Main.Operation | Specifies the operation of the virtual channel | uint8 | 25c1 | 9665 | Not applicable |
| VirtualChannel.25.Main.Period | The time period over which the calculation is made | int32 | 25ca | 9674 | Not applicable |
| VirtualChannel.25.Main.Preset | Initiate preset. 0 = No; 1 = Yes | bool | 25cc | 9676 | Not applicable |
| VirtualChannel.25.Main.PresetValue | Specifies the preset value | float32 | 25cd | 9677 | Set by VirtualChannel.25.Main.Resolution |
| VirtualChannel.25.Main.PV | The virtual channel output value | float32 | 016e | 366 | Set by VirtualChannel.25.Main.Resolution |
| VirtualChannel.25.Main.Reset | Initiate reset. 0 = No; 1 = Yes | bool | 25cb | 9675 | Not applicable |
| VirtualChannel.25.Main.Resolution | Specifies the resolution/number of decimal places | uint8 | 25c2 | 9666 | Not applicable |
| VirtualChannel.25.Main.Rollover | A pulse signal to indicate PV (output) has just rolled over | bool | 25d1 | 9681 | Not applicable |
| VirtualChannel.25.Main.RolloverValue | Rollover value | float32 | 25d2 | 9682 | Set by VirtualChannel.25.Main.Resolution |
| VirtualChannel.25.Main.Status | As VirtualChannel1.Main.Status | uint8 | 016f | 367 | Not applicable |
| VirtualChannel.25.Main.TimeRemaining | Time remaining before the calculation is made | time_t | 25c9 | 9673 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.25.Main.Trigger | Increment/decrement counter. 0 = No; 1 = Yes | bool | 25ce | 9678 | Not applicable |
| VirtualChannel.25.Main.Type | As VirtualChannel1.Main.Type | uint8 | 25c0 | 9664 | Not applicable |
| VirtualChannel.25.Main.Units | Units descriptor | string_t | 4d9e | 19870 | Not applicable |
| VirtualChannel.25.Main.UnitsScaler | Units scaler for totalisers | float32 | 25c3 | 9667 | 1dp |
| VirtualChannel.25.Trend.Colour | As VirtualChannel1.Trend.Colour | uint8 | 26e0 | 9952 | Not applicable |
| VirtualChannel.25.Trend.SpanHigh | Specifies the highest PV (output value) to be displayed | float32 | 26e2 | 9954 | Same as VirtualChannel.25.Main.PV |
| VirtualChannel.25.Trend.SpanLow | Specifies the lowest PV (output value) to be displayed | float32 | 26e1 | 9953 | Same as VirtualChannel.25.Main.PV |
| VirtualChannel.26.Main.Descriptor | Virtual Channel descriptor | string_t | 4da4 | 19876 | Not applicable |
| VirtualChannel.26.Main.Disable | 1 = Virtual channel disabled | bool | 2723 | 10019 | Not applicable |
| VirtualChannel.26.Main.HighCutOff | The highest input value that will be totalised/counted | float32 | 2605 | 9733 | Set by VirtualChannel.26.Main.Resolution |
| VirtualChannel.26.Main.Input1 | Input 1 value | float32 | 2607 | 9735 | Set by VirtualChannel.26.Main.Resolution |
| VirtualChannel.26.Main.Input2 | Input 2 value | float32 | 2608 | 9736 | Set by VirtualChannel.26.Main.Resolution |
| VirtualChannel.26.Main.LowCutOff | The lowest input value that will be totalised/counted | float32 | 2604 | 9732 | Set by VirtualChannel.26.Main.Resolution |
| VirtualChannel.26.Main.ModbusInput | Modbus input value | float32 | 2606 | 9734 | Set by VirtualChannel.26.Main.Resolution |
| VirtualChannel.26.Main.Operation | Specifies the operation of the virtual channel | uint8 | 2601 | 9729 | Not applicable |
| VirtualChannel.26.Main.Period | The time period over which the calculation is made | int32 | 260a | 9738 | Not applicable |
| VirtualChannel.26.Main.Preset | Initiate preset. 0 = No; 1 = Yes | bool | 260c | 9740 | Not applicable |
| VirtualChannel.26.Main.PresetValue | Specifies the preset value | float32 | 260d | 9741 | Set by VirtualChannel.26.Main.Resolution |
| VirtualChannel.26.Main.PV | The virtual channel output value | float32 | 0170 | 368 | Set by VirtualChannel.26.Main.Resolution |
| VirtualChannel.26.Main.Reset | Initiate reset. 0 = No; 1 = Yes | bool | 260b | 9739 | Not applicable |
| VirtualChannel.26.Main.Resolution | Specifies the resolution/number of decimal places | uint8 | 2602 | 9730 | Not applicable |
| VirtualChannel.26.Main.Rollover | A pulse signal to indicate PV (output) has just rolled over | bool | 2611 | 9745 | Not applicable |
| VirtualChannel.26.Main.RolloverValue | Rollover value | float32 | 2612 | 9746 | Set by VirtualChannel.26.Main.Resolution |
| VirtualChannel.26.Main.Status | As VirtualChannel1.Main.Status | uint8 | 0171 | 369 | Not applicable |
| VirtualChannel.26.Main.TimeRemaining | Time remaining before the calculation is made | time_t | 2609 | 9737 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.26.Main.Trigger | Increment/decrement counter. 0 = No; 1 = Yes | bool | 260e | 9742 | Not applicable |
| VirtualChannel.26.Main.Type | As VirtualChannel1.Main.Type | uint8 | 2600 | 9728 | Not applicable |
| VirtualChannel.26.Main.Units | Units descriptor | string_t | 4db9 | 19897 | Not applicable |
| VirtualChannel.26.Main.UnitsScaler | Units scaler for totalisers | float32 | 2603 | 9731 | 1dp |
| VirtualChannel.26.Trend.Colour | As VirtualChannel1.Trend.Colour | uint8 | 2720 | 10016 | Not applicable |
| VirtualChannel.26.Trend.SpanHigh | Specifies the highest PV (output value) to be displayed | float32 | 2722 | 10018 | Same as VirtualChannel.26.Main.PV |
| VirtualChannel.26.Trend.SpanLow | Specifies the lowest PV (output value) to be displayed | float32 | 2721 | 10017 | Same as VirtualChannel.26.Main.PV |
| VirtualChannel.27.Main.Descriptor | Virtual Channel descriptor | string_t | 4dbf | 19903 | Not applicable |
| VirtualChannel.27.Main.Disable | 1 = Virtual channel disabled | bool | 2763 | 10083 | Not applicable |
| VirtualChannel.27.Main.HighCutOff | The highest input value that will be totalised/counted | float32 | 2645 | 9797 | Set by VirtualChannel.27.Main.Resolution |
| VirtualChannel.27.Main.Input1 | Input 1 value | float32 | 2647 | 9799 | Set by VirtualChannel.27.Main.Resolution |
| VirtualChannel.27.Main.Input2 | Input 2 value | float32 | 2648 | 9800 | Set by VirtualChannel.27.Main.Resolution |
| VirtualChannel.27.Main.LowCutOff | The lowest input value that will be totalised/counted | float32 | 2644 | 9796 | Set by VirtualChannel.27.Main.Resolution |
| VirtualChannel.27.Main.ModbusInput | Modbus input value | float32 | 2646 | 9798 | Set by VirtualChannel.27.Main.Resolution |
| VirtualChannel.27.Main.Operation | Specifies the operation of the virtual channel | uint8 | 2641 | 9793 | Not applicable |
| VirtualChannel.27.Main.Period | The time period over which the calculation is made | int32 | 264a | 9802 | Not applicable |
| VirtualChannel.27.Main.Preset | Initiate preset. 0 = No; 1 = Yes | bool | 264c | 9804 | Not applicable |
| VirtualChannel.27.Main.PresetValue | Specifies the preset value | float32 | 264d | 9805 | Set by VirtualChannel.27.Main.Resolution |
| VirtualChannel.27.Main.PV | The virtual channel output value | float32 | 0172 | 370 | Set by VirtualChannel.27.Main.Resolution |
| VirtualChannel.27.Main.Reset | Initiate reset. 0 = No; 1 = Yes | bool | 264b | 9803 | Not applicable |
| VirtualChannel.27.Main.Resolution | Specifies the resolution/number of decimal places | uint8 | 2642 | 9794 | Not applicable |
| VirtualChannel.27.Main.Rollover | A pulse signal to indicate PV (output) has just rolled over | bool | 2651 | 9809 | Not applicable |
| VirtualChannel.27.Main.RolloverValue | Rollover value | float32 | 2652 | 9810 | Set by VirtualChannel.27.Main.Resolution |
| VirtualChannel.27.Main.Status | As VirtualChannel1.Main.Status | uint8 | 0173 | 371 | Not applicable |
| VirtualChannel.27.Main.TimeRemaining | Time remaining before the calculation is made | time_t | 2649 | 9801 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.27.Main.Trigger | Increment/decrement counter. 0 = No; 1 = Yes | bool | 264e | 9806 | Not applicable |
| VirtualChannel.27.Main.Type | As VirtualChannel1.Main.Type | uint8 | 2640 | 9792 | Not applicable |
| VirtualChannel.27.Main.Units | Units descriptor | string_t | 4dd4 | 19924 | Not applicable |
| VirtualChannel.27.Main.UnitsScaler | Units scaler for totalisers | float32 | 2643 | 9795 | 1dp |
| VirtualChannel.27.Trend.Colour | As VirtualChannel1.Trend.Colour | uint8 | 2760 | 10080 | Not applicable |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| VirtualChannel.27.Trend.SpanHigh | Specifies the highest PV (output value) to be displayed | float32 | 2762 | 10082 | Same as VirtualChannel.27.Main.PV |
| VirtualChannel.27.Trend.SpanLow | Specifies the lowest PV (output value) to be displayed | float32 | 2761 | 10081 | Same as VirtualChannel.27.Main.PV |
| VirtualChannel.28.Main.Descriptor | Virtual Channel descriptor | string_t | 4dda | 19930 | Not applicable |
| VirtualChannel.28.Main.Disable | 1 = Virtual channel disabled | bool | 27a3 | 10147 | Not applicable |
| VirtualChannel.28.Main.HighCutOff | The highest input value that will be totalised/counted | float32 | 2685 | 9861 | Set by VirtualChannel.28.Main.Resolution |
| VirtualChannel.28.Main.Input1 | Input 1 value | float32 | 2687 | 9863 | Set by VirtualChannel.28.Main.Resolution |
| VirtualChannel.28.Main.Input2 | Input 2 value | float32 | 2688 | 9864 | Set by VirtualChannel.28.Main.Resolution |
| VirtualChannel.28.Main.LowCutOff | The lowest input value that will be totalised/counted | float32 | 2684 | 9860 | Set by VirtualChannel.28.Main.Resolution |
| VirtualChannel.28.Main.ModbusInput | Modbus input value | float32 | 2686 | 9862 | Set by VirtualChannel.28.Main.Resolution |
| VirtualChannel.28.Main.Operation | Specifies the operation of the virtual channel | uint8 | 2681 | 9857 | Not applicable |
| VirtualChannel.28.Main.Period | The time period over which the calculation is made | int32 | 268a | 9866 | Not applicable |
| VirtualChannel.28.Main.Preset | Initiate preset. 0 = No; 1 = Yes | bool | 268c | 9868 | Not applicable |
| VirtualChannel.28.Main.PresetValue | Specifies the preset value | float32 | 268d | 9869 | Set by VirtualChannel.28.Main.Resolution |
| VirtualChannel.28.Main.PV | The virtual channel output value | float32 | 0174 | 372 | Set by VirtualChannel.28.Main.Resolution |
| VirtualChannel.28.Main.Reset | Initiate reset. 0 = No; 1 = Yes | bool | 268b | 9867 | Not applicable |
| VirtualChannel.28.Main.Resolution | Specifies the resolution/number of decimal places | uint8 | 2682 | 9858 | Not applicable |
| VirtualChannel.28.Main.Rollover | A pulse signal to indicate PV (output) has just rolled over | bool | 2691 | 9873 | Not applicable |
| VirtualChannel.28.Main.RolloverValue | Rollover value | float32 | 2692 | 9874 | Set by VirtualChannel.28.Main.Resolution |
| VirtualChannel.28.Main.Status | As VirtualChannel1.Main.Status | uint8 | 0175 | 373 | Not applicable |
| VirtualChannel.28.Main.TimeRemaining | Time remaining before the calculation is made | time_t | 2689 | 9865 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.28.Main.Trigger | Increment/decrement counter. 0 = No; 1 = Yes | bool | 268e | 9870 | Not applicable |
| VirtualChannel.28.Main.Type | As VirtualChannel1.Main.Type | uint8 | 2680 | 9856 | Not applicable |
| VirtualChannel.28.Main.Units | Units descriptor | string_t | 4def | 19951 | Not applicable |
| VirtualChannel.28.Main.UnitsScaler | Units scaler for totalisers | float32 | 2683 | 9859 | 1dp |
| VirtualChannel.28.Trend.Colour | As VirtualChannel1.Trend.Colour | uint8 | 27a0 | 10144 | Not applicable |
| VirtualChannel.28.Trend.SpanHigh | Specifies the highest PV (output value) to be displayed | float32 | 27a2 | 10146 | Same as VirtualChannel.28.Main.PV |
| VirtualChannel.28.Trend.SpanLow | Specifies the lowest PV (output value) to be displayed | float32 | 27a1 | 10145 | Same as VirtualChannel.28.Main.PV |
| VirtualChannel.29.Main.Descriptor | Virtual Channel descriptor | string_t | 4df5 | 19957 | Not applicable |
| VirtualChannel.29.Main.Disable | 1 = Virtual channel disabled | bool | 27e3 | 10211 | Not applicable |
| VirtualChannel.29.Main.HighCutOff | The highest input value that will be totalised/counted | float32 | 26c5 | 9925 | Set by VirtualChannel.29.Main.Resolution |
| VirtualChannel.29.Main.Input1 | Input 1 value | float32 | 26c7 | 9927 | Set by VirtualChannel.29.Main.Resolution |
| VirtualChannel.29.Main.Input2 | Input 2 value | float32 | 26c8 | 9928 | Set by VirtualChannel.29.Main.Resolution |
| VirtualChannel.29.Main.LowCutOff | The lowest input value that will be totalised/counted | float32 | 26c4 | 9924 | Set by VirtualChannel.29.Main.Resolution |
| VirtualChannel.29.Main.ModbusInput | Modbus input value | float32 | 26c6 | 9926 | Set by VirtualChannel.29.Main.Resolution |
| VirtualChannel.29.Main.Operation | Specifies the operation of the virtual channel | uint8 | 26c1 | 9921 | Not applicable |
| VirtualChannel.29.Main.Period | The time period over which the calculation is made | int32 | 26ca | 9930 | Not applicable |
| VirtualChannel.29.Main.Preset | Initiate preset. 0 = No; 1 = Yes | bool | 26cc | 9932 | Not applicable |
| VirtualChannel.29.Main.PresetValue | Specifies the preset value | float32 | 26cd | 9933 | Set by VirtualChannel.29.Main.Resolution |
| VirtualChannel.29.Main.PV | The virtual channel output value | float32 | 0176 | 374 | Set by VirtualChannel.29.Main.Resolution |
| VirtualChannel.29.Main.Reset | Initiate reset. 0 = No; 1 = Yes | bool | 26cb | 9931 | Not applicable |
| VirtualChannel.29.Main.Resolution | Specifies the resolution/number of decimal places | uint8 | 26c2 | 9922 | Not applicable |
| VirtualChannel.29.Main.Rollover | A pulse signal to indicate PV (output) has just rolled over | bool | 26d1 | 9937 | Not applicable |
| VirtualChannel.29.Main.RolloverValue | Rollover value | float32 | 26d2 | 9938 | Set by VirtualChannel.29.Main.Resolution |
| VirtualChannel.29.Main.Status | As VirtualChannel1.Main.Status | uint8 | 0177 | 375 | Not applicable |
| VirtualChannel.29.Main.TimeRemaining | Time remaining before the calculation is made | time_t | 26c9 | 9929 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.29.Main.Trigger | Increment/decrement counter. 0 = No; 1 = Yes | bool | 26ce | 9934 | Not applicable |
| VirtualChannel.29.Main.Type | As VirtualChannel1.Main.Type | uint8 | 26c0 | 9920 | Not applicable |
| VirtualChannel.29.Main.Units | Units descriptor | string_t | 4e0a | 19978 | Not applicable |
| VirtualChannel.29.Main.UnitsScaler | Units scaler for totalisers | float32 | 26c3 | 9923 | 1dp |
| VirtualChannel.29.Trend.Colour | As VirtualChannel1.Trend.Colour | uint8 | 27e0 | 10208 | Not applicable |
| VirtualChannel.29.Trend.SpanHigh | Specifies the highest PV (output value) to be displayed | float32 | 27e2 | 10210 | Same as VirtualChannel.29.Main.PV |
| VirtualChannel.29.Trend.SpanLow | Specifies the lowest PV (output value) to be displayed | float32 | 27e1 | 10209 | Same as VirtualChannel.29.Main.PV |
| VirtualChannel.30.Main.Descriptor | Virtual Channel descriptor | string_t | 4e10 | 19984 | Not applicable |
| VirtualChannel.30.Main.Disable | 1 = Virtual channel disabled | bool | 2823 | 10275 | Not applicable |
| VirtualChannel.30.Main.HighCutOff | The highest input value that will be totalised/counted | float32 | 2705 | 9989 | Set by VirtualChannel.30.Main.Resolution |
| VirtualChannel.30.Main.Input1 | Input 1 value | float32 | 2707 | 9991 | Set by VirtualChannel.30.Main.Resolution |
| VirtualChannel.30.Main.Input2 | Input 2 value | float32 | 2708 | 9992 | Set by VirtualChannel.30.Main.Resolution |
| VirtualChannel.30.Main.LowCutOff | The lowest input value that will be totalised/counted | float32 | 2704 | 9988 | Set by VirtualChannel.30.Main.Resolution |
| VirtualChannel.30.Main.ModbusInput | Modbus input value | float32 | 2706 | 9990 | Set by VirtualChannel.30.Main.Resolution |
| VirtualChannel.30.Main.Operation | Specifies the operation of the virtual channel | uint8 | 2701 | 9985 | Not applicable |
| VirtualChannel.30.Main.Period | The time period over which the calculation is made | int32 | 270a | 9994 | Not applicable |
| VirtualChannel.30.Main.Preset | Initiate preset. 0 = No; 1 = Yes | bool | 270c | 9996 | Not applicable |
| VirtualChannel.30.Main.PresetValue | Specifies the preset value | float32 | 270d | 9997 | Set by VirtualChannel.30.Main.Resolution |
| VirtualChannel.30.Main.PV | The virtual channel output value | float32 | 0178 | 376 | Set by VirtualChannel.30.Main.Resolution |
| VirtualChannel.30.Main.Reset | Initiate reset. 0 = No; 1 = Yes | bool | 270b | 9995 | Not applicable |
| VirtualChannel.30.Main.Resolution | Specifies the resolution/number of decimal places | uint8 | 2702 | 9986 | Not applicable |
| VirtualChannel.30.Main.Rollover | A pulse signal to indicate PV (output) has just rolled over | bool | 2711 | 10001 | Not applicable |
| VirtualChannel.30.Main.RolloverValue | Rollover value | float32 | 2712 | 10002 | Set by VirtualChannel.30.Main.Resolution |
| VirtualChannel.30.Main.Status | As VirtualChannel1.Main.Status | uint8 | 0179 | 377 | Not applicable |
| VirtualChannel.30.Main.TimeRemaining | Time remaining before the calculation is made | time_t | 2709 | 9993 | Set by Network.Modbus.TimeFormat |
| VirtualChannel.30.Main.Trigger | Increment/decrement counter. 0 = No; 1 = Yes | bool | 270e | 9998 | Not applicable |
| VirtualChannel.30.Main.Type | As VirtualChannel1.Main.Type | uint8 | 2700 | 9984 | Not applicable |
| VirtualChannel.30.Main.Units | Units descriptor | string_t | 4e25 | 20005 | Not applicable |
| VirtualChannel.30.Main.UnitsScaler | Units scaler for totalisers | float32 | 2703 | 9987 | 1dp |
| VirtualChannel.30.Trend.Colour | As VirtualChannel1.Trend.Colour | uint8 | 2820 | 10272 | Not applicable |
| VirtualChannel.30.Trend.SpanHigh | Specifies the highest PV (output value) to be displayed | float32 | 2822 | 10274 | Same as VirtualChannel.30.Main.PV |
| VirtualChannel.30.Trend.SpanLow | Specifies the lowest PV (output value) to be displayed | float32 | 2821 | 10273 | Same as VirtualChannel.30.Main.PV |
| Zirconia.aC_CO_O2 | Carbon Activity Between CO and O2 | float32 | 289e | 10398 | 4dp |

## 5.3 PARAMETER LIST (Cont.)

| Parameter path | Description | Type | Hex | Dec | Resolution |
|---|---|---|---|---|---|
| Zirconia.BalanceIntegral | Balance Integral | bool | 289d | 10397 | Not applicable |
| Zirconia.CarbonPot | Calculated Carbon Potential | float32 | 2892 | 10386 | Set by Zirconia.Resolution |
| Zirconia.Clean.AbortClean | 1 = Abort cleaning process | bool | 28b5 | 10421 | Not applicable |
| Zirconia.Clean.CantClean | 1 = can't clean | bool | 28c3 | 10435 | Not applicable |
| Zirconia.Clean.CleanAbort | 1 = Cleaning cycle has been aborted | bool | 28c4 | 10436 | Not applicable |
| Zirconia.Clean.CleanEnable | 1 = probe cleaning allowed | bool | 28b2 | 10418 | Not applicable |
| Zirconia.Clean.CleanFreq | Interval between probe cleaning cycles | time_t | 28aa | 10410 | Set by Network.Modbus.TimeFormat |
| Zirconia.Clean.CleanMaxTemp | Maximum temperature for cleaning. If, during the cleaning cycle, the probe temperature exceeds this value, cleaning is aborted. | float32 | 28b4 | 10420 | 0dp |
| Zirconia.Clean.CleanMsgReset | 1 = Clear cleaning related alarms | bool | 28b3 | 10419 | Not applicable |
| Zirconia.Clean.CleanProbe | 1 = Initiate a probe cleaning cycle | bool | 28b0 | 10416 | Not applicable |
| Zirconia.Clean.CleanRecoveryTime | The time taken to recover from last clean.<br>0 = max. clean recovery time exceeded last time | time_t | 28b6 | 10422 | Set by Network.Modbus.TimeFormat |
| Zirconia.Clean.CleanTemp | 1 = Clean cycle aborted because cleaning temperature was too high. | bool | 28c5 | 10437 | Not applicable |
| Zirconia.Clean.CleanTime | The time for which the probe is cleaned | time_t | 28ab | 10411 | Set by Network.Modbus.TimeFormat |
| Zirconia.Clean.CleanValve | 1 = Enable probe cleaning valve | bool | 28af | 10415 | Not applicable |
| Zirconia.Clean.LastCleanMv | Probe output after last clean, in mV | float32 | 28b7 | 10423 | 0dp |
| Zirconia.Clean.MaxRcovTime | Max. recovery time after a purge | time_t | 28ad | 10413 | Set by Network.Modbus.TimeFormat |
| Zirconia.Clean.MinRcovTime | Min. recovery time after a purge | time_t | 28ac | 10412 | Set by Network.Modbus.TimeFormat |
| Zirconia.Clean.ProbeFault | 1 = Probe failed to recover following the clean cycle | bool | 28ae | 10414 | Not applicable |
| Zirconia.Clean.Time2Clean | Time to next cleaning cycle | time_t | 28b1 | 10417 | Set by Network.Modbus.TimeFormat |
| Zirconia.CleanFreq | Interval between cleaning cycles | time_t | 2889 | 10377 | Set by Network.Modbus.TimeFormat |
| Zirconia.CleanProbe | Initiates a demand cleaning cycle | bool | 289a | 10394 | Not applicable |
| Zirconia.CleanState | Cleaning State (0 = Waiting, 1 = Cleaning, 2 = Recovering) | uint8 | 2899 | 10393 | Not applicable |
| Zirconia.CleanTime | The time for which the probe is cleaned | time_t | 288a | 10378 | Set by Network.Modbus.TimeFormat |
| Zirconia.CleanValve | 1 = Enable probe cleaning valve | bool | 2898 | 10392 | Not applicable |
| Zirconia.DewPoint | Calculated Dewpoint | float32 | 2893 | 10387 | Set by Zirconia.Resolution |
| Zirconia.GasRef | Reference value for hydrogen concentration | float32 | 2882 | 10370 | 1dp |
| Zirconia.GasRefs.CO_Ideal | Gas ref value when Oxygen Type = Nernst | float32 | 28a9 | 10409 | 1dp |
| Zirconia.GasRefs.CO_InUse | The CO gas measurement value being used | float32 | 28a4 | 10404 | 1dp |
| Zirconia.GasRefs.CO_Local | Reference value for CO concentration | float32 | 28a1 | 10401 | 1dp |
| Zirconia.GasRefs.CO_Remote | CO concentration from remote source | float32 | 28a2 | 10402 | 1dp |
| Zirconia.GasRefs.CO_RemoteEn | 1 = Allow remote gas measurement | bool | 28a3 | 10403 | Not applicable |
| Zirconia.GasRefs.H2_InUse | The hydrogen gas measurement value being used | float32 | 28a8 | 10408 | 1dp |
| Zirconia.GasRefs.H2_Local | Reference value for hydrogen concentration | float32 | 28a5 | 10405 | 1dp |
| Zirconia.GasRefs.H2_Remote | Hydrogen concentration from remote source | float32 | 28a6 | 10406 | 1dp |
| Zirconia.GasRefs.H2_RemoteEn | 1 = Allow remote gas measurement | bool | 28a7 | 10407 | Not applicable |
| Zirconia.MaxRcovTime | Maximum recovery time after a purge | time_t | 288c | 10380 | Set by Network.Modbus.TimeFormat |
| Zirconia.MinCalTemp | Min. temp at which the calculation is valid | float32 | 2886 | 10374 | Same as Zirconia.TempInput |
| Zirconia.MinRcovTime | Minimum recovery time after a purge | time_t | 288b | 10379 | Set by Network.Modbus.TimeFormat |
| Zirconia.NumResolution | Number of decimal places | uint8 | 2881 | 10369 | Not applicable |
| Zirconia.Oxygen | Calculated Oxygen value | float32 | 2894 | 10388 | Set by Zirconia.Resolution |
| Zirconia.OxygenExp | Exponent used by log oxygen calculations | int16 | 288d | 10381 | Not applicable |
| Zirconia.OxygenType | The oxygen equation being used.uint8<br>0 = Nernst 1=Nernst Bosch<br>2 = Nernst CP 3= Ferronova | 28a0 | | 10400 | Not applicable |
| Zirconia.ProbeFault | Probe Clean Recovery Warning | bool | 2896 | 10390 | Not applicable |
| Zirconia.ProbeInput | Probe input in mV | float32 | 2890 | 10384 | 0dp |
| Zirconia.ProbeOffset | Probe offset in mV | float32 | 2891 | 10385 | Set by Zirconia.Resolution |
| Zirconia.ProbeState | State of the probe measurement system<br>0 = Measuring 1 = Cleaning<br>2 = Clean Recovery 3 = Test impedance<br>4 = Impedance recovery 5 = Not ready | uint8 | 289f | 10399 | Not applicable |
| Zirconia.ProbeStatus | Status of Probe uint8<br>0 = OK 1 = mVSbr<br>2 = TempSbr 3 = MincalcT | 289c | | 10396 | Not applicable |
| Zirconia.ProbeType | Type of Probe<br>25 = MMI<br>26 = AACC 27 = Dray 28 = Accu<br>29 = SSI 30 = MacD 31 = Bosch<br>32 = Barber 33 = ferono 34 = PrbmV<br>35 = Eurotherm | uint8 | 2880 | 10368 | Not applicable |
| Zirconia.ProcFactor | Process Factor (Value defined by probe manufacturer) | float32 | 2888 | 10376 | 1dp |
| Zirconia.PVFrozen | 1 = PV frozen | bool | 2897 | 10391 | Not applicable |
| Zirconia.RemGasEn | 1 = Enable use of remote gas reference | bool | 2884 | 10372 | Not applicable |
| Zirconia.RemGasRef | Remote Gas Reference Value | float32 | 2883 | 10371 | 1dp |
| Zirconia.SootAlm | 1 = Soot alarm active | bool | 2895 | 10389 | Not applicable |
| Zirconia.TempInput | Probe temperature Input | float32 | 288e | 10382 | 0dp |
| Zirconia.TempOffset | Temperature Offset | float32 | 288f | 10383 | Set by Zirconia.Resolution |
| Zirconia.Time2Clean | Time To Next Clean | time_t | 289b | 10395 | Set by Network.Modbus.TimeFormat |
| Zirconia.Tolerance | Sooting Tolerance | float32 | 2887 | 10375 | 1dp |
| Zirconia.WrkGas | Working Reference Gas Value | float32 | 2885 | 10373 | 1dp |

# 6 iTOOLS

iTools software running on a pc allows quick and easy access to the configuration of the unit. The parameters used are generally the same as those described in section 4 above, with the addition of various diagnostic parameters.

iTools also gives the user the ability to create software wiring between function blocks, such wiring being carried out using the Graphical wiring Editor feature.

A further feature - the display mode 'Promote List', is populated using iTools - see section 3.4.11 for details.

In addition to the guidance given here, there are two on-line Help systems available within iTools: Parameter help and iTools help. Parameter help is accessed by clicking on 'Help' in the toolbar (opens the complete parameter help system), by right-clicking on a parameter and selecting 'Parameter Help' from the resulting context menu, or by clicking on the Help menu and selecting 'Device Help'. iTools help is accessed by clicking on the Help menu, and selecting 'Contents'. iTools help is also available in manual format under part number HA028838, either as a physical manual or as a pdf file.

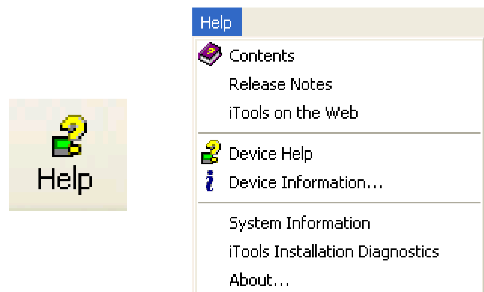



Figure 6 Help access

## 6.1 iTools CONNECTION

The following descriptions assume that iTools software has been correctly installed on the pc.

### 6.1.1 Ethernet (Modbus TCP) communications

Note: the following description is based on windows XP. Windows 'Vista' is similar.

It is first necessary to determine the IP address of the unit, as described under 'Network.Interface' in section 4.2.1.

Once the Ethernet link has been correctly installed, carry out the following actions at the pc:

1. Click on 'Start'
2. Click on 'Control Panel'. (If Control Panel opens in 'Category View' select 'Classic View' instead.)
3. Double-click on 'iTools'.
4. Click on the TCP/IP tab in the Registry settings configuration.
5. Click on 'Add...' The 'New TCP/IP Port' dialogue box opens.
6. Type-in a name for the port, then click 'Add...' again
7. Type the IP address of the unit in the 'Edit Host' box which appears. Click OK.
8. Check the details in the 'New TCP/IP Port' box, then click on 'OK'.
9. Click on 'OK' in the 'Registry settings' box to confirm the new port.

(Continued)

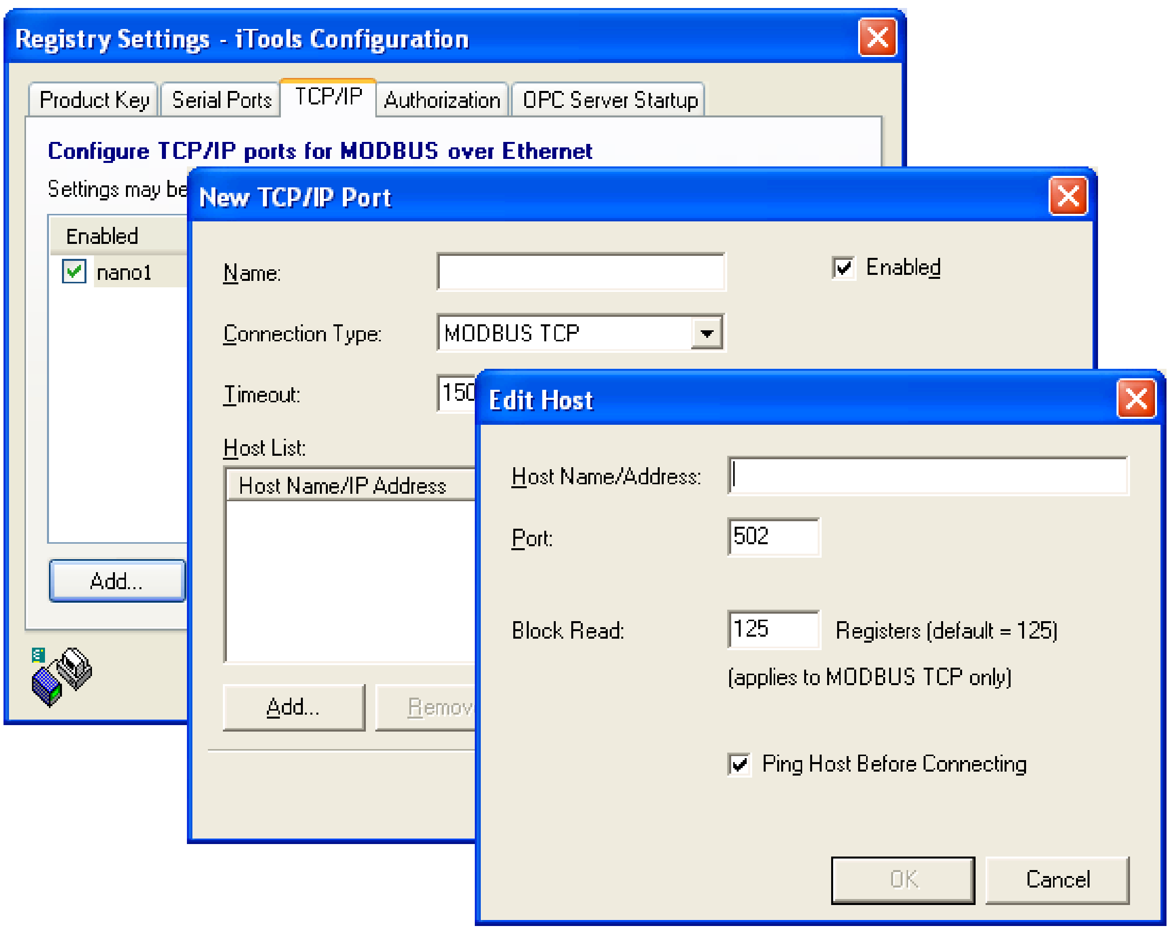


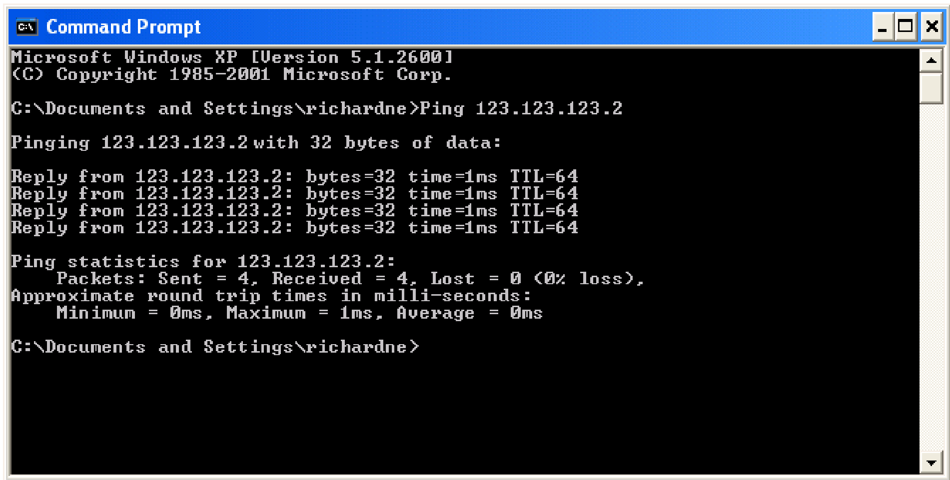

Figure 6.1.1a Adding a new Ethernet port

## 6.1.1 ETHERNET (TCP/IP) COMMUNICATIONS (Cont.)

To check that the pc can now communicate with the instrument, Click 'Start'. 'All Programs', 'Accessories', 'Command Prompt'

when the Command Prompt box appears, type in: Ping<Space>IP1.IP2.IP3.IP4<Enter> (where IP1 to IP4 are the IP address of the instrument).

If the Ethernet link to the instrument is operating correctly, the 'successful' reply arrives. Otherwise, the 'failed' reply arrives, in which case, the Ethernet link, IP address, and pc port details should be verified.



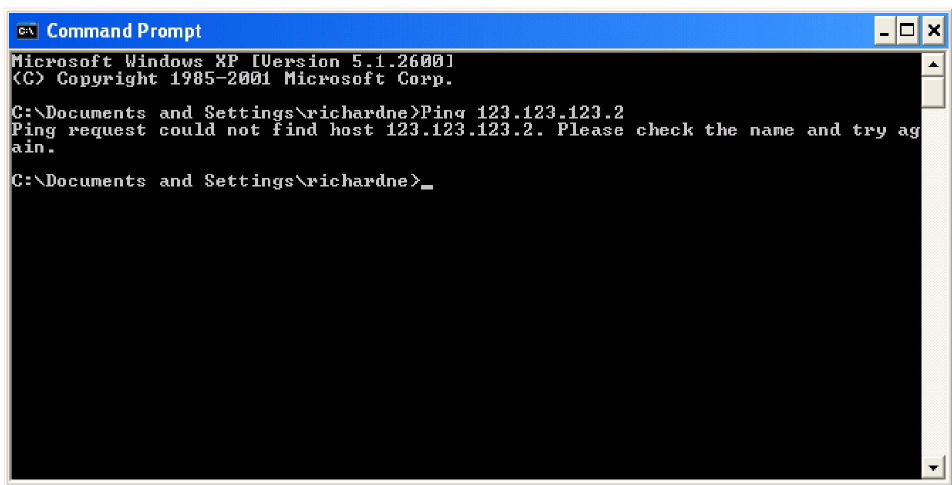
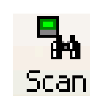




Figure 6.1.1b Command prompt 'Ping' screens (typical)

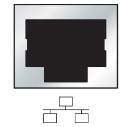
Once the Ethernet link to the instrument has been verified, iTools can be started (or shut down and restarted), and the Scan toolbar icon used, to 'find' the instrument. The scan can be stopped at any time by clicking on the Scan icon a second time.




See section 6.2 for more details of the scan procedure.

### 6.1.2 Direct Connection

This section describes how to connect a pc directly to the instrument.



PC Ethernet connector.

**WIRING**

Connection is made from the Ethernet connector at the rear of the Instrument to an Ethernet RJ45 connector, usually located at the rear of the pc. The cable can be either a 'cross-over' or 'straight through' type.

Once wired correctly, and powered up, it is necessary to enter a suitable IP address and subnet mask into the Comms configuration of the Driver Module. This information can be found as follows:

1. At the pc, click 'Start'. 'All Programs', 'Accessories', 'Command Prompt'
2. When the Command Prompt box appears, type IPConfig<Enter>

   The response is a display, such as that shown below, giving the IP address and Subnet mask of the pc. Choose an address in the range covered by these two values.

   A subnet mask element of 255 means that the equivalent element of the IP address must be used unchanged. A subnet mask element of 0 means that the equivalent element of the IP address may take any value between 1 and 255 (0 is not allowed). In the example below, the range of IP addresses which may be chosen for the Driver Module is 123.123.123.2 to 123.123.123.255. (123.123.123.0 is not allowed and 123.123.123.1 is the same as the pc's address, and may therefore not be used.)

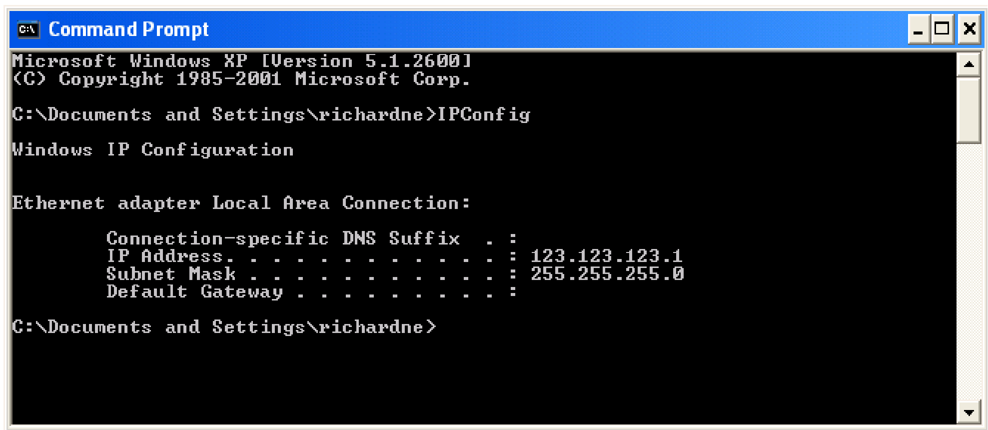



Figure 6.1.2b IP Config command

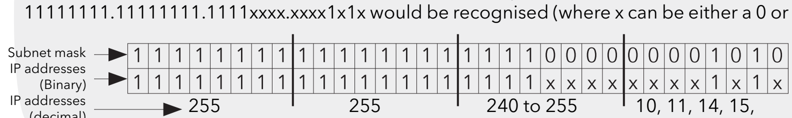
3. In Network.Interface configuration (section 4.2.1) enter the selected IP address and the subnet mask (as it appears in the command prompt window) in the relevant parts of the menu.
4. Check communications by 'pinging' as described in section 6.1.1, above.

Once the link to the instrument has been verified, iTools can be started (or shut down and re-started), and the Scan toolbar icon used, to 'find' the instrument. The scan can be stopped at any time by clicking on the Scan icon a second time.

See section 6.2 for more details of the scan procedure.

**Subnet Masks and IP addresses.**

Subnet Masks are most readily understood when looked at in binary format.
For example, a mask of 255.255.240.10 can be re-written as:
11111111.11111111.11110000.00001010. In such a case, IP addresses
11111111.11111111.1111xxxx.xxxx1x1x would be recognised (where x can be either a 0 or



## 6.2 SCANNING FOR INSTRUMENTS

Clicking on the 'Scan' toolbar icon causes a dialogue box (shown below) to appear. This allows the user to define a search range of addresses.

Notes:

1. The relevant instrument address is that entered in the Network.Modbus configuration item (section 4.2.4, and it can take any value between 1 and 254 inclusive, as long as it is unique to the comms link.
2. The default selection (Scan all device addresses...) will detect any instrument on the serial link, which has a valid address.

As the search progresses, any instruments detected by the scan appear as thumbnails (faceplates) in the 'Panel Views' area, normally located at the bottom of the iTools screen. (options/Panel Views position allows this area to be moved to the top of the window, or the Close icon can be used to close it. Once closed it can be re-opened by clicking on 'Panel Views' in the 'View' menu.)

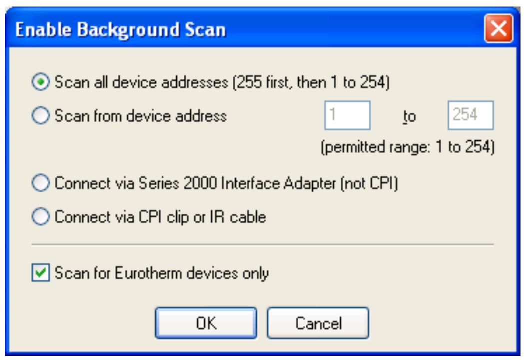



Figure 6.2a Scan range enable

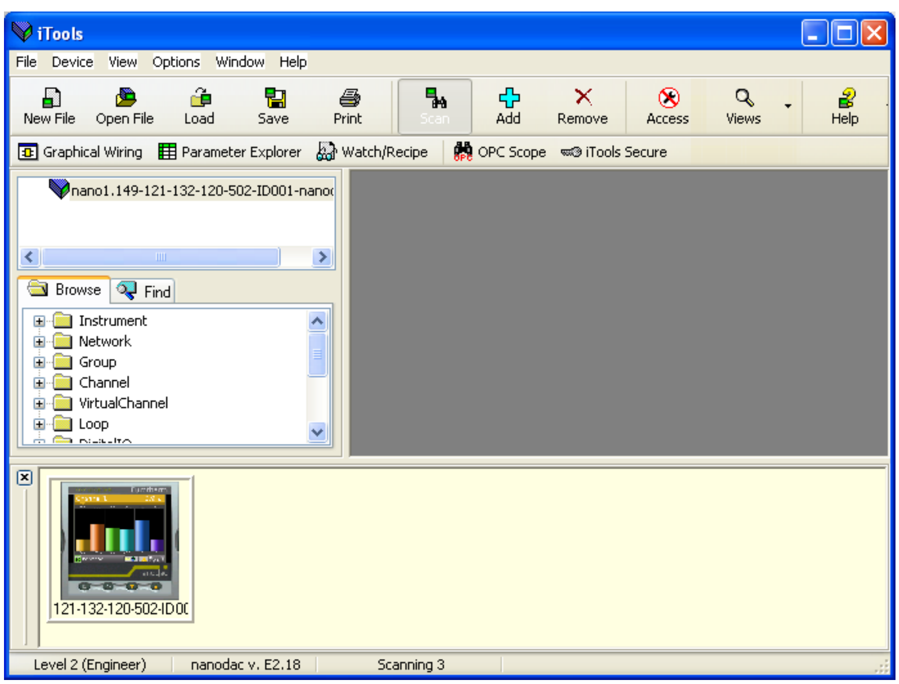



Figure 6.2b iTools initial window with one instrument detected

Once the instrument has been detected stop the scan. When the instrument has synchronised, click on the 'Access' button to enter configuration mode (a password might be required). Once the editing session is complete, click on the Access button again to quit configuration mode.

## 6.3 GRAPHICAL WIRING EDITOR 


Clicking on the Graphical wiring Editor tool bar icon causes the Graphical wiring window for the current instrument configuration to open.

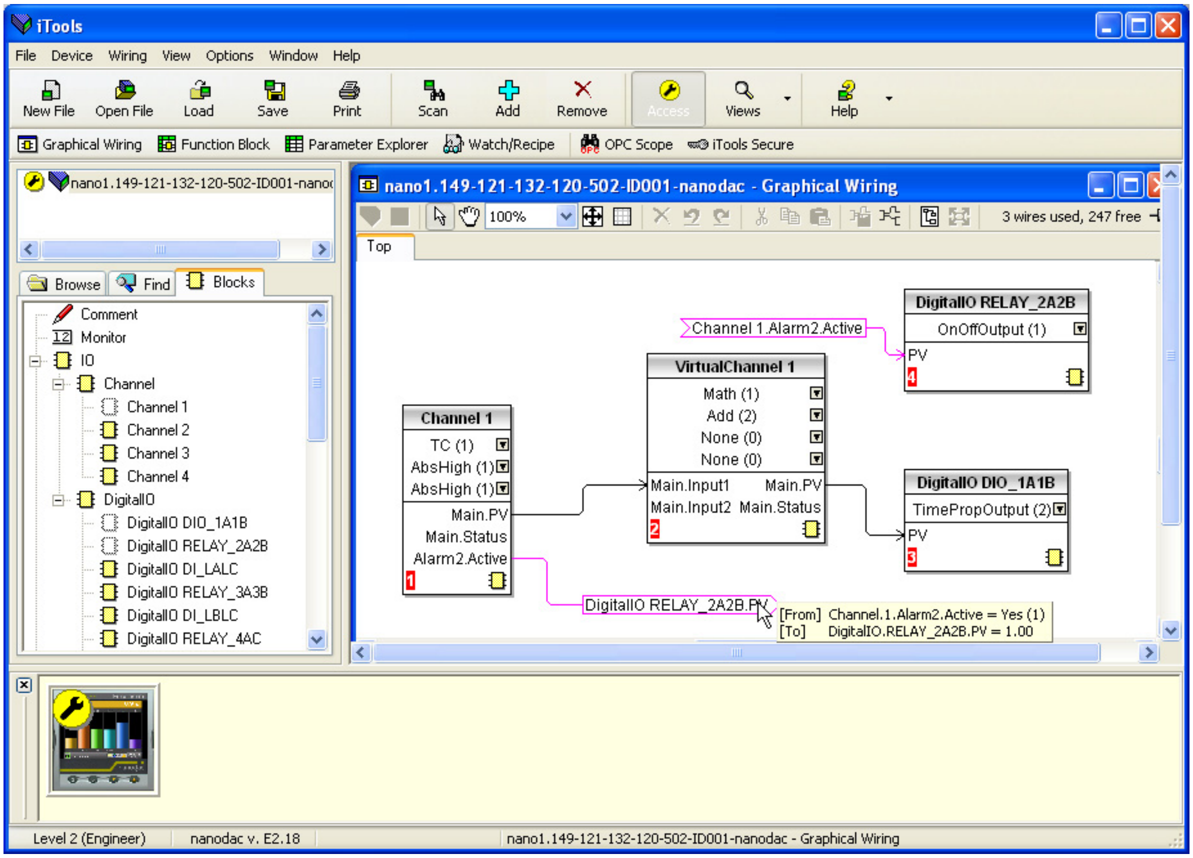



Figure 6.3 Graphical wiring Editor

The graphical wiring editor allows:

1. Function blocks, notes, comments etc. to be 'drag and dropped' into the wiring diagram from the tree list (left pane).
2. Parameters to be wired to one another by clicking on the output, the clicking on the required input.
3. Viewing and/or editing of parameter values by right-clicking on a function block and selecting 'Function Block View'.
4. The user to select parameter lists and to switch between parameter and wiring editors.
5. Completed wiring to be downloaded to the instrument (function blocks and wiring items with dashed outlines are new, or have been edited since the last download).

### 6.3.1 Tool bar


 Download wiring to instrument.

 Mouse select. Select normal mouse operation. Mutually exclusive with ‘Mouse Pan’ below.

 Mouse Pan. When active, this causes the mouse cursor to change to a hand-shaped icon. Allows the graphical wiring diagram to be click-dragged within the GWE window aperture.

Zoom. Allows the magnification factor of the wiring diagram to be selected.

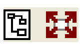
 Pan tool. Whilst left clicked, the cursor appears as a rectangle showing which part of the wiring diagram is currently displayed. Click dragging allows the rectangle to be moved freely about the diagram. The size of the rectangle depends on the zoom setting.

 Show/Hide grid. This toggles an alignment grid on and off.

 Undo, redo.Allows the user to undo the last action, or, once an undo action has taken place, to undo the undo. Short cuts are <Ctrl>+<Z. for undo; <Ctrl>+<V, for undo.

 Cut, Copy, Paste. Normal Cut (copy and delete), Copy (copy without delete) and Paste (insert into) functions. Shortcuts are: <Ctrl> + <X> for ‘Cut’; <Ctrl> + <C> for copy and <Ctrl> + <V> for Paste.

 Copy diagram fragment; Paste diagram fragment. Allows a part of the wiring diagram to be selected, named and be saved to file. The fragment may then be pasted into any wiring diagram, including the source diagram.

 Create compound; Flatten compound. These two icons allow compounds to be created and ‘un created’ (flattened).

### 6.3.2 Wiring editor operating details

#### COMPONENT SELECTION

Single wires are shown with boxes at ‘corners’ when selected. When more than one wire is selected, as part of a group, the wire colour changes to magenta. All other items have a dashed line drawn round them when selected.

Clicking on a single item selects it. An Item can be added to the selection by holding down the control key (ctrl) whilst clicking on the item. (A selected item can be deselected in the same way.) If a block is selected, then all its associated wires are also selected.

Alternatively, the mouse can be click-dragged on the background to create a ‘rubber band’ round the relevant area; anything within this area being selected when the mouse is released.

<Ctrl>+<A> selects all items on the active diagram.

#### BLOCK EXECUTION ORDER

The order in which the blocks are executed by the instrument depends on the way in which they are wired. Each block displays its place in its sequence in a coloured block in the bottom left-hand corner (figure 6.3.2a).

## 6.3.2 WIRING EDITOR OPERATING DETAILS (Cont.)

### FUNCTION BLOCKS

A Function Block is an algorithm which may be wired to and from other function blocks to make a control strategy. Each function block has inputs and outputs. Any parameter may be wired from, but only parameters that are alterable in Operator Mode may we wired to. A function block includes any parameters that are needed to configure or operate the algorithm. The inputs and outputs which are considered to be of most use are always shown. In most cases all of these need to be wired before the block can perform a useful task.

If a function block is not faded in the tree (left hand pane) it can be dragged onto the diagram. The block can be dragged around the diagram using the mouse.

A Channel block is shown below as an example. When block type information is alterable (as in this case) click on the box with the down arrow in it to display a dialogue box allowing the value to be edited.

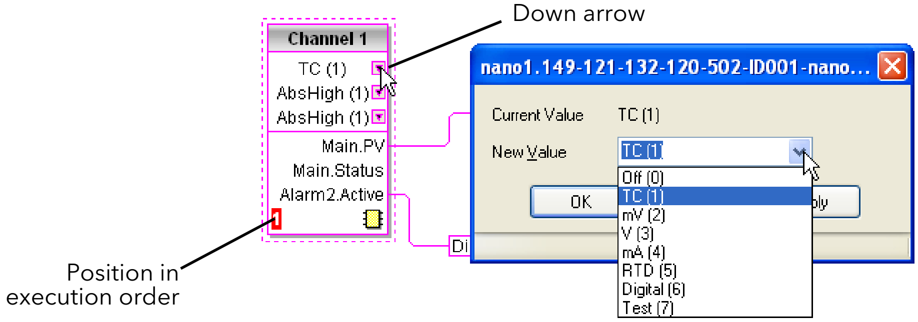
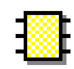



Figure 6.3.2a Function block example

If it is required to wire from a parameter, which is not shown as a recommended output, click on the 'Click to Select Output' icon in the bottom right hand corner to display a full list of parameters in the block (figure 6.3.2c, below). Click on one of these to start a wire.



FUNCTION BLOCK CONTEXT MENU

Right click in the function block to display the context menu.

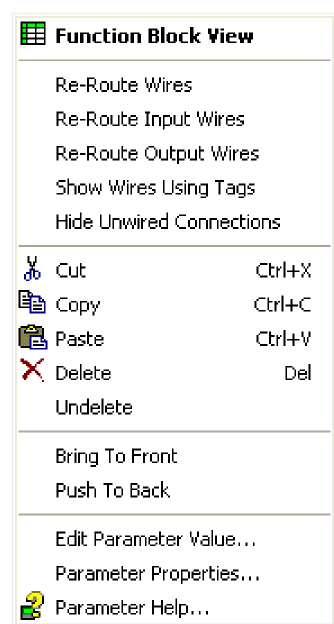



Figure 6.3.2b Function block context menu

Function block view Displays a list of parameters associated with the function block. 'Hidden' parameters can be displayed by de-selecting 'Hide Parameters and Lists when not Relevant in the options menu 'Parameter availability setting...' item

Re-Route wires Redraws all wiring associated with the function block.

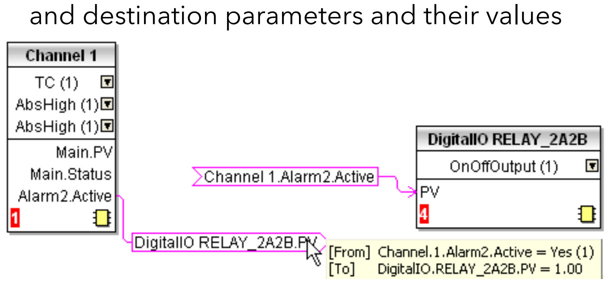
Re-route input wires Redraws all input wiring associated with the function block

Re-route output wires Redraws all output wiring associated with the function block.

Show wiring using tags

Wires are not drawn, but their start and end destinations are indicated by tags instead. Reduces wire clutter in diagrams where source and destination are widely separated.

Hovering the cursor over the tag shows both its source and destination parameters and their values



## 6.3.2 WIRING EDITOR OPERATING DETAILS (Cont.)

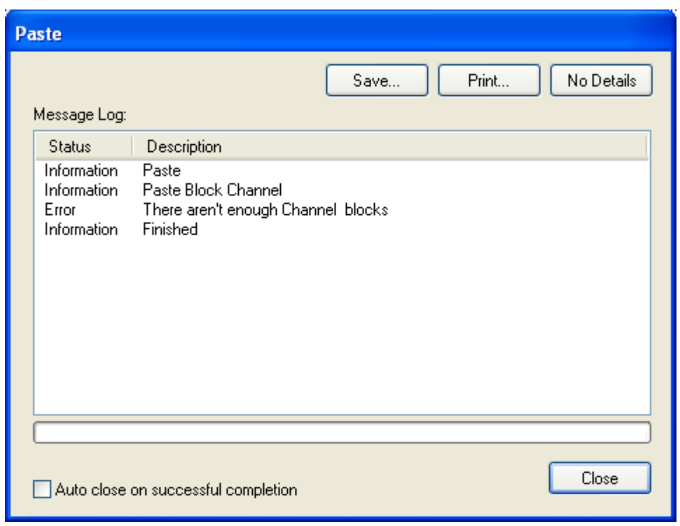
### FUNCTION BLOCK CONTEXT MENU (Cont.)

Hide unwanted connections
: Causes the display to include only wired items.

Cut
: Allows one or more selected items to be moved to the Clipboard ready for pasting into another diagram or compound, or for use in a Watch window, or OPC scope. The original items are greyed out, and function blocks and wires are shown dashed until next download, after which they are removed from the diagram. Short cut = <Ctrl>+<X>. Cut operations carried out since the last download can be 'undone' by using the 'Undo' tool bar icon, by selecting 'Undelete' or by using the short cut <Ctrl>+<Z>.

Copy
: Allows one or more selected items to be copied to the Clipboard ready for pasting into another diagram or compound, or for use in a Watch window, or OPC scope. The original items remain in the current wiring diagram. Short cut = <Ctrl>+<C>. If items are pasted to the same diagram from which they were copied, the items will be replicated with different block instances. Should this result in more instances of a block than are available, an error display appears showing details of which items couldn't be copied.

Paste
: Copies items from the Clipboard to the current wiring diagram. Short cut = <Ctrl>+<V>. If items are pasted to the same diagram from which they were copied, the items will be replicated with different block instances. Should this result in more instances of a block than are available, a Paste error display appears showing details of those items which could not be copied.




Delete
: Marks all selected items for deletion. Such items are shown dashed until next download, after which they are removed from the diagram. Short cut = <Del>.

Undelete
: Reverses 'Delete' and 'Cut' operations carried out on selected item(s) since the last download.

Bring to Front
: Brings selected items to the front of the diagram.

Push to Back
: Sends the selected items to the back of the diagram.

Edit Parameter Value...
: This menu item is active if the cursor is hovering over an editable parameter. Selecting this menu item causes a pop-up window to appear, which allows the user to edit the parameter value.

Parameter Properties
: This menu item is active if the cursor is hovering over an editable parameter. Selecting this menu item causes a pop-up window to appear, which allows the user to view the parameter properties, and also, to view the parameter Help (by clicking on the 'Help' tab).

Parameter Help
: Produces Parameter Properties and Help information for the selected function block or parameter, depending on the hover position of the cursor, when the right-click occurs.

## 6.3.2 WIRING EDITOR OPERATING DETAILS (Cont.)

### WIRES

To make a wire

1. Drag two (or more) blocks onto the diagram from the function block tree.
2. Start a wire by either clicking on a recommended output or clicking on the 'Click to Select output' icon at the bottom right corner of the block to bring up the connection dialogue, and clicking on the required parameter. Recommended connections are shown with a green plug symbol; other parameters which are available being shown in yellow. Clicking on the red button causes all parameters to be shown. To dismiss the connection dialogue either press the escape key on the keyboard, or click the cross at the bottom left of the dialogue box.
3. Once the wire has started a dashed wire is drawn from the output to the current mouse position. To complete the wire click on the required destination parameter.
4. Wires remain dashed until they are downloaded

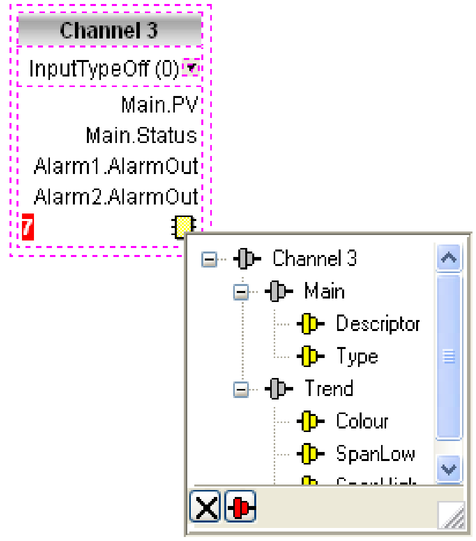



Figure 6.3.2c Output selection dialogue box.

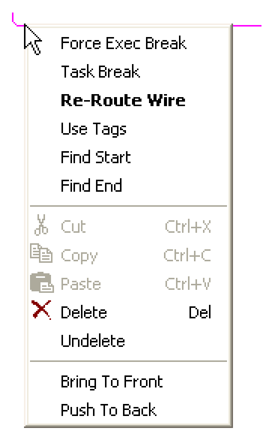
Routing wires

When a wire is placed it is auto-routed. The auto routing algorithm searches for a clear path between the two blocks. A wire can be auto-routed again using the context menus or by double clicking the wire. A wire segment can be edited manually by click-dragging. If the block to which it is connected is moved, the end of the wire moves with it, retaining as much of the path as possible.

If a wire is selected by clicking on it, it is drawn with small boxes on its corners.

Wire Context Menu

Right click on a wire to display the wire block context menu:




| | |
|---|---|
| Force Exec Break | When wires form a loop, a break point must be introduced, where the value written to the block comes from a source which was last executed during the previous cycle. A break is automatically placed by iTools, and appears in red. Force Exec Break allows the user to define where a break must be placed. Surplus breaks appear in black. |
| Re-Route wire | Replaces the current wire route with a new route generated from scratch. |
| Use Tags | Toggles between wire and tag mode between parameters. Tag mode is useful for sources and destinations which are widely separated. |
| Find Start | Goes to the source of the wire. |
| Find End | Goes to the destination of the wire. |
| Cut, Copy, Paste | Not used in this context. |
| Delete | Marks the wire for deletion. The wire is redrawn as a dashed line (or dashed tags) until next download. Operation can be reversed until after next download. |
| Undelete | Reverses the effect of the Delete operation up until the next download, after which, Undelete is disabled. |
| Bring to Front | Brings the wire to the front of the diagram. |
| Push to Back | Sends the wire to the back of the diagram. |

### 6.3.2 WIRING EDITOR OPERATING DETAILS (Cont.)

Wire Colours

| | |
|---|---|
| Black | Normal functioning wire |
| Red | The wire is connected to a non-changeable parameter. Values are rejected by the destination block. |
| Magenta | A normal functioning wire is being hovered-over by the mouse cursor. |
| Purple | A red wire is being hovered-over by the mouse cursor. |
| Green | New Wire (dashed green wire changes to solid black after being downloaded.) |

**COMMENTS**

Comments are added to a wiring diagram by click-dragging them from the Function Block tree onto the diagram. As soon as the mouse is released, a dialogue box opens to allow the comment text to be entered.

Carriage returns are used to control the width of the comment. Once text entry is complete, 'OK' causes the comment to appear on the diagram. There are no restrictions on the size of a comment. Comments are saved to the instrument along with the diagram layout information.

Comments can be linked to function blocks and wires by clicking on the chain icon at the bottom right-hand corner of the comment box and then clicking again on the required block or wire. A dashed line is drawn to the top of the block or to the selected wire segment (figure 6.3.2f).

> Note: Once the comment has been linked, the Chain icon disappears. It re-appears when the mouse cursor is hovered over the bottom right-hand corner of the comment box.

Comment Context Menu

| | |
|---|---|
| Edit | Opens the Comment dialogue box to allow the comment text to be edited. |
| Unlink | Deletes the current link from the comment. |
| Cut | Moves the comment to the Clipboard, ready to be pasted elsewhere. Short cut = <Ctrl>+<X>. |
| Copy | Copies the comment from the wiring diagram to the Clipboard, ready to be pasted elsewhere. Short cut = <Ctrl>+<C>. |
| Paste | Copies a comment from the Clipboard to the wiring diagram. Short cut = <Ctrl>+<V>. |
| Delete | Marks the comment for deletion at next download. |
| Undelete | Undoes the Delete command if download has not taken place since. |

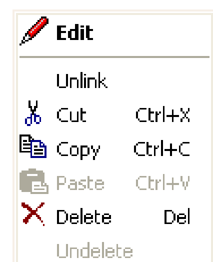



Figure 6.3.2e
Comment context menu

### 6.3.2 WIRING EDITOR OPERATING DETAILS (Cont.)

#### MONITORS

Monitor points are added to a wiring diagram by click-dragging them from the Function Block tree onto the diagram. A monitor shows the current value (updated at the iTools parameter list update rate) of the parameter to which it is linked. By default the name of the parameter is shown. To hide the parameter name either double click on the monitor box or 'Show Names' in the context (right-click) menu can be used to toggle the parameter name on and off.

Monitors are linked to function blocks and wires by clicking on the chain icon at the bottom right-hand corner of the box and then clicking again on the required parameter. A dashed line is drawn to the top of the block or the selected wire segment.

Note: Once the monitor has been linked, the Chain icon disappears. It re-appears when the mouse cursor is hovered over the bottom right-hand corner of the monitor box.

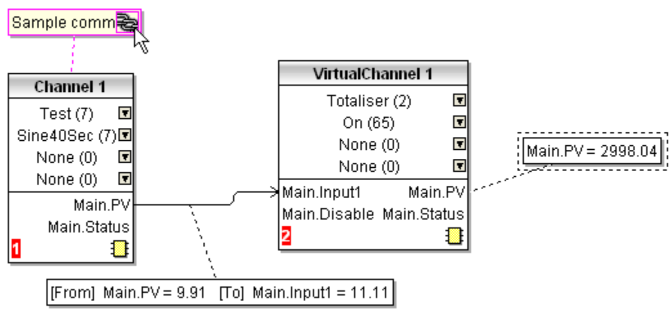



Figure 6.3.2f Comment and Monitor appearance

Monitor Context Menu

| | |
|---|---|
| Show names | Toggles parameter names on and off in the monitor box. |
| Unlink | Deletes the current link from the monitor. |
| Cut | Moves the monitor to the Clipboard, ready to be pasted elsewhere. Short cut = <Ctrl>+<X>. |
| Copy | Copies the monitor from the wiring diagram to the Clipboard, ready to be pasted elsewhere. Short cut = <Ctrl>+<C>. |
| Paste | Copies a monitor from the Clipboard to the wiring diagram. Short cut = <Ctrl>+<V>. |
| Delete | Marks the monitor for deletion at next download. |
| Undelete | Undoes the Delete command if download has not taken place since. |
| Bring to Front | Moves the item to the 'top' layer of the diagram. |
| Push to Back | Moves the item to the 'bottom' layer of the diagram. |
| Parameter Help | Shows parameter help for the item. |

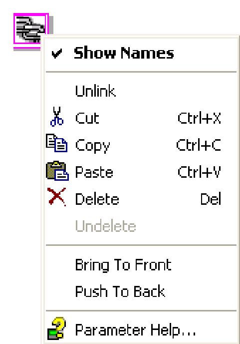



Figure 6.3.2g
Monitor context menu

#### DOWNLOADING

When the wiring editor is opened the current wiring and diagram layout is read from the instrument. No changes are made to the instrument function block execution or wiring until the download button is pressed. Any changes made using the operator interface after the editor is opened are lost on download.

When a block is dropped onto the diagram, instrument parameters are changed to make the parameters for that block available. If changes are made and the editor is closed without saving them there is a delay while the editor clears these parameters.

During download, the wiring is written to the instrument which then calculates the block execution order and starts executing the blocks. The diagram layout including comments and monitors is then written into instrument flash memory along with the current editor settings. When the editor is reopened, the diagram is shown positioned as it was when it was last downloaded.

## 6.3.2 WIRING EDITOR OPERATING DETAILS (Cont.)

### COLOURS

Items on the diagram are coloured as follows:

| | |
|---|---|
| Red | Items which totally or partially obscure other items and items which are totally or partially obscured by other items. Wires that are connected to unalterable or non-available parameters. Execution breaks. |
| Blue | Non-available parameters in function blocks. |
| Green | Items added to the diagram since last download are shown as green dashed lines. |
| Magenta | All selected items, or any item over which the cursor is hovering. |
| Purple | Red wires when being hovered over by the mouse cursor. |
| Black | All items added to the diagram before the last download. Redundant execution breaks. Monitor and comment text. |

### DIAGRAM CONTEXT MENU

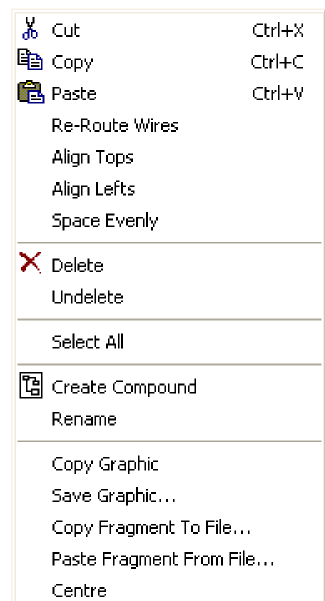


Figure 6.3.2h
Diagram context menu

| | |
|---|---|
| Cut | Active only when the right click occurs within the bounding rectangle which appears when more than one item is selected. Moves the selection off the diagram to the Clipboard. Short cut = <Ctrl>+<X>. |
| Copy | As for 'Cut', but the selection is copied, leaving the original on the diagram. Short cut = <Ctrl>+<C>. |
| Paste | Copies the contents of the Clipboard to the diagram. Short cut = <Ctrl>+<V>. |
| Re-Route wires | Reroutes all selected wires. If no wires are selected, all wires are re-routed. |
| Align Tops | Aligns the tops of all blocks in the selected area. |
| Align Lefts | Aligns the left edges of all blocks in the selected area. |
| Space Evenly | Spaces selected items such that their top left corners are spaced evenly across the width of the diagram. Click on the item which is to be the left-most item, then <Ctrl>+<left click> the remaining items in the order in which they are to appear. |
| Delete | Marks the item for deletion at next download time. Can be 'Undeleted' up until download occurs. |
| Undelete | Reverses the action of 'Delete' on the selected item. |
| Select All | Selects all items on the current diagram. |
| Create Compound | Active only when the right click occurs, in the top level diagram, within the bounding rectangle which appears when more than one item is selected. Creates a new wiring diagram as described in 'Compound', below. |
| Rename | Allows a new name to entered for the current wiring diagram. This name appears in the relevant tab. |
| Copy Graphic | Copies the selected items (or the whole diagram if no items are selected) to the clipboard as a Windows metafile, suitable for pasting into a documentation application. Wiring entering/leaving the selection (if any) are drawn in tag mode. |
| Save Graphic... | As for 'Copy Graphic' above, but saves to a user-specified file location instead of the clipboard. |
| Copy Fragment To File... | Copies selected items to a user-named file in folder 'My iTools Wiring Fragments' located in 'My Documents'. |
| Paste Fragment From File | Allows the user to select a stored fragment for inclusion in the wiring diagram. |
| Centre | Places the display window at the centre of the selected items. If 'Select All' has previously been clicked-on, then the display widow is placed over the centre of the diagram. |

### 6.3.2 WIRING EDITOR OPERATING DETAILS (Cont.)

### COMPOUNDS

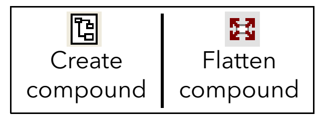
Compounds are used to simplify the top level wiring diagram, by allowing the placing of any number of function blocks within one 'box', the inputs and outputs of which operate in the same way as those of a normal function block.

Each time a compound is created, a new tab appears at the top of the wiring diagram. Initially compounds and their tabs are named 'Compound 1', 'Compound 2', etc. but they can be renamed by right clicking either on the compound in the top level diagram, or anywhere within an open Compound, selecting 'Rename' and typing in the required text string (16 characters max.).

Compounds cannot contain other compounds (i.e. they can be created only in the top level diagram).

Compound creation

1. Empty compounds are created within the top level diagram by clicking on the 'Create Compound' tool bar icon.
2. Compounds can also be created by highlighting one or more function blocks in the top level diagram and then clicking on the 'Create Compound' tool bar icon. The highlighted items are moved from the top level diagram into a new compound.




3. Compounds are 'uncreated' (flattened), by highlighting the relevant item in the top level menu and then clicking on the 'Flatten Compound' tool bar icon. All the items previously contained within the compound appear on the top level diagram.
4. Wiring between top level and compound parameters is carried out by clicking on the source parameter, then clicking on the compound (or the compound tab) and then clicking on the destination parameter. Wiring from a compound parameter to a top level parameter or from compound to compound is carried out in similar manner.
5. Unused function blocks can be moved into compounds by dragging from the tree view. Existing blocks can be dragged from the top level diagram, or from another compound, onto the tab associated with the destination compound. Blocks are moved out of compounds to the top level diagram or to another compound in a similar way. Function blocks can also be 'cut and pasted'.
6. Default compound names (e.g. 'Compound 2') are used only once, so that if, for example, Compounds 1 and 2 have been created, and Compound 2 is subsequently deleted, then the next compound to be created will be named 'Compound 3'.
7. Top level elements can be click-dragged into compounds.

### TOOL TIPS

Hovering the cursor over the block displays 'tooltips' describing that part of the block beneath the cursor. For function block parameters the tooltip shows the parameter description, its OPC name, and, if downloaded, its value. Similar tooltips are shown when hovering over inputs, outputs and over many other items on the iTools screen.

A Function Block is enabled by dragging the block onto the diagram, wiring it, and finally downloading it to the instrument. Initially blocks and associated wires are drawn with dashed lines, and when in this state the parameter list for the block is enabled but the block is not executed by the instrument.

The block is added to the instrument function block execution list when the 'Download' icon is operated and the items are redrawn using solid lines.

If a block which has been downloaded is deleted, it is shown on the diagram in a ghosted form until the download button is pressed. (This is because it and any wires to/from it are still being executed in the instrument. On download it will be removed from the instrument execution list and the diagram.) A ghosted block can be 'undeleted' as described in 'Context menu', above.

When a dashed block is deleted it is removed immediately.

## 6.4 PARAMETER EXPLORER


This view can be displayed:

1. by clicking on the 'Parameter Explorer' toolbar icon,
2. by double clicking on the relevant block in the tree pane or in the graphical wiring editor
3. by selecting 'Function Block View' from the Function block context menu in the Graphical wiring Editor.
4. by selecting 'parameter Explorer from the 'View' menu
5. by using the short cut <Alt>+<Enter>

In each case the function block parameters appear in the iTools window in tabular form, such as the example in figure 6.4a, below.

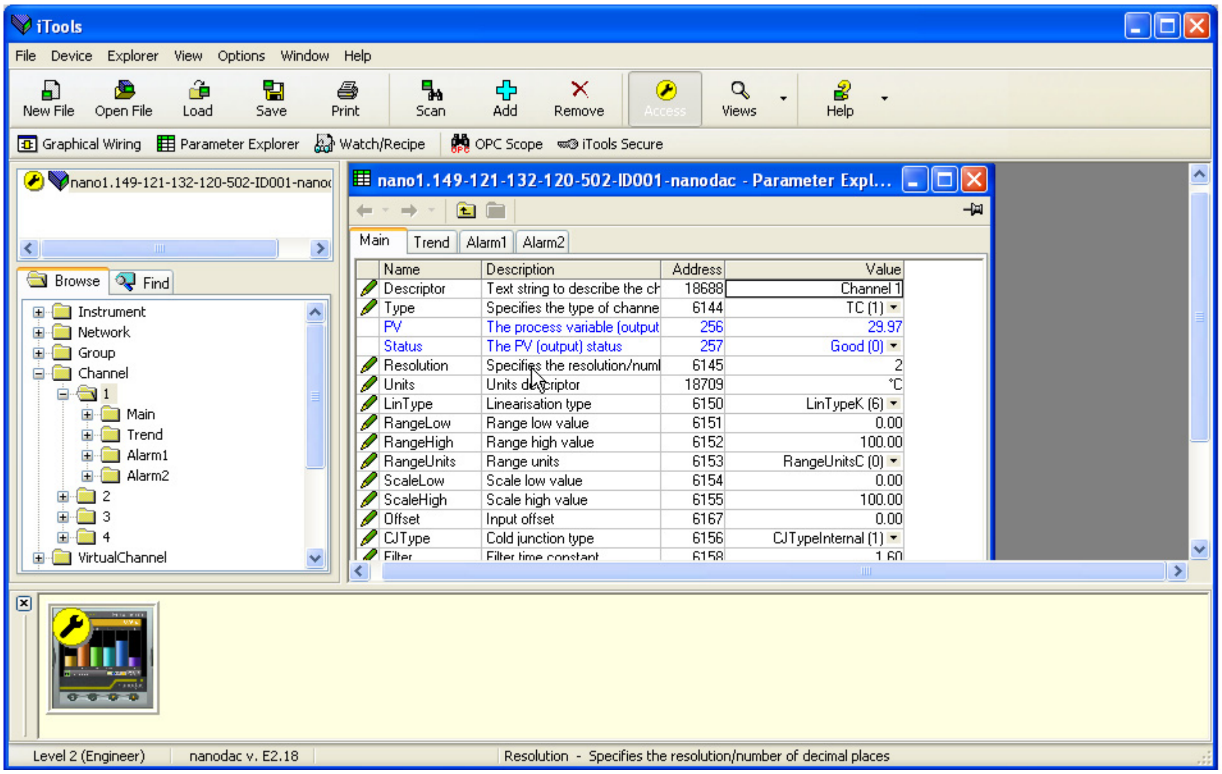



Figure 6.4a Parameter table example

The figure above shows the default table layout. Columns can be added/deleted from the view using the 'Columns' item of the Explorer or context menus (figure 6.4b).

## 6.4 PARAMETER EXPLORER (Cont.)

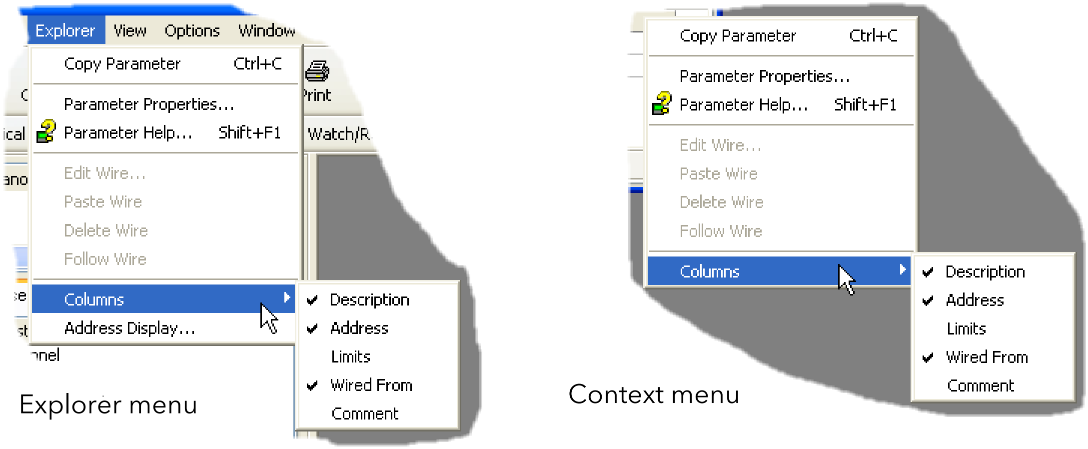
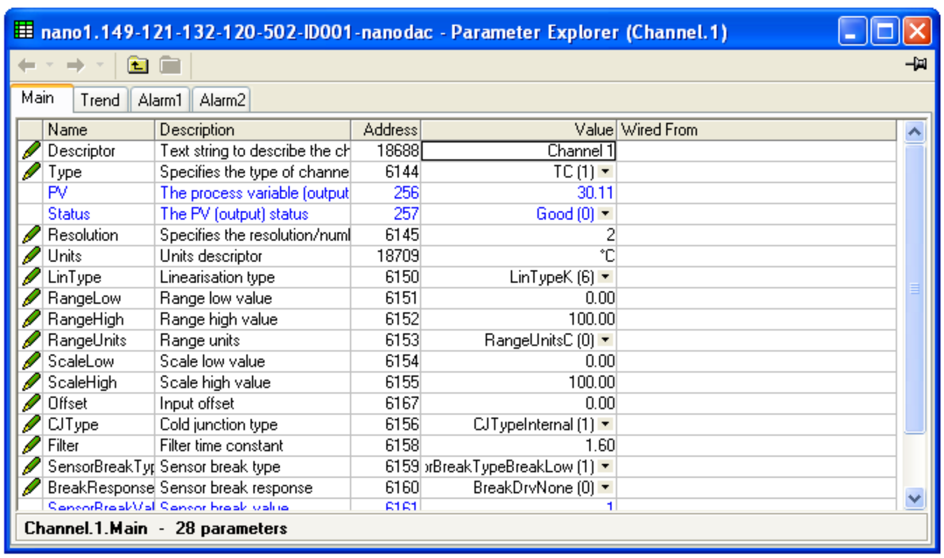



Figure 6.4b Column enable/disable

### 6.4.1 Parameter explorer detail

Figure 6.4.1a shows a typical parameter table. This particular parameter has a number of subfolders associated with it, and each of these is represented by a 'tab' across the top of the table.




| Name | Description | Address | Value | Wired From |
|---|---|---|---|---|
| Descriptor | Text string to describe the ch | 18688 | Channel 1 | |
| Type | Specifies the type of channe | 6144 | TC (1) | |
| PV | The process variable (output | 256 | 30.11 | |
| Status | The PV (output) status | 257 | Good (0) | |
| Resolution | Specifies the resolution/numl | 6145 | 2 | |
| Units | Units descriptor | 18709 | °C | |
| LinType | Linearisation type | 6150 | LinTypeK (6) | |
| RangeLow | Range low value | 6151 | 0.00 | |
| RangeHigh | Range high value | 6152 | 100.00 | |
| RangeUnits | Range units | 6153 | RangeUnitsC (0) | |
| ScaleLow | Scale low value | 6154 | 0.00 | |
| ScaleHigh | Scale high value | 6155 | 100.00 | |
| Offset | Input offset | 6167 | 0.00 | |
| CJType | Cold junction type | 6156 | CJTypeInternal (1) | |
| Filter | Filter time constant | 6158 | 1.60 | |
| SensorBreakTyp | Sensor break type | 6159 | rBreakTypeBreakLow (1) | |
| BreakResponse | Sensor break response | 6160 | BreakDrvNone (0) | |
| SensorBreakVal | Sensor break value | 6161 | 1 | |



Figure 6.4.1a Typical parameter table

Notes:

1. Parameters in blue are non-editable (Read only). In the example above all the parameters are read only. Read/write parameters are in black and have a 'pencil' symbol in the 'read/Write access column at the left edge of the table. A number of such items are shown in figure 6.4.1a, above.
2. Columns. The default explorer window (figure 6.4a) contains the columns 'Name', 'Description', 'Address', 'Value', and 'Wired From'. As can be seen from figure 6.4b, the columns to be displayed can be selected, to a certain extent, using either the 'Explorer' menu or the context menu. 'Limits' have been enabled for the example above.
3. Hidden Parameters. By default, iTools hides parameters which are considered irrelevant in the current context. Such hidden parameters can be shown in the table using the 'Parameter availability' settings item of the options menu (figure 6.4.1b). Such items are displayed with a shaded background.
4. The full pathname for the displayed parameter list is shown at the bottom left hand corner of the window.

### 6.4.1 PARAMETER EXPLORER DETAIL (Cont.)

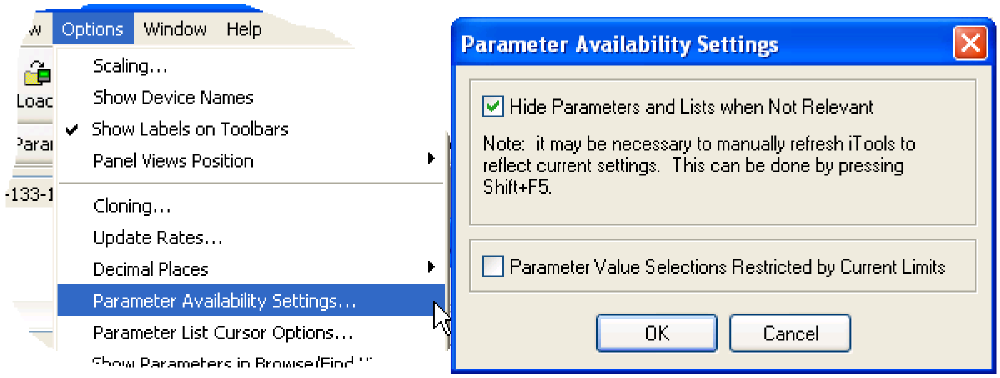
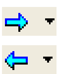



Figure 6.4.1b Show/Hide parameters

## 6.4.2 Explorer tools

A number of tool icons appear above the parameter list:

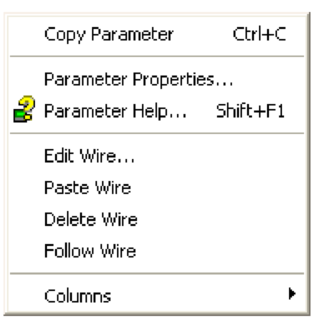


Back to: and Forward to:. The parameter explorer contains a history buffer of up to 10 lists that have been browsed in the current instance of the window. The 'Back to: (list name)' and 'Forward to: (list name)' icons allow easy retracing or repeating of the parameter list view sequence.
If the mouse cursor is hovered over the tool icon, the name of the parameter list which will appear if the icon is clicked-on appears. Clicking on the arrow head displays a pick list of up to 10 previously visited lists which the user can select. Short cut = <Ctrl>+<B> for 'Back to' or <Ctrl>+<F> for 'Forward to'.

Go Up a Level, Go Down a Level. For nested parameters, these buttons allow the user to navigate 'vertically' between levels. Short cut = <Ctrl>+<U> for 'Go Up a Level' or <Ctrl>+<D> for 'Go Down a Level'.

Push pin to give the window global scope. Clicking on this icon causes the current parameter list to be permanently displayed, even if another instrument becomes the 'current device'.

## 6.4.3 Context Menu




Copy Parameter Copies the clicked-on parameter to the clipboard

Parameter properties
Displays parameter properties for the clicked-on parameter

Parameter Help... Displays help information for the clicked-on parameter

Edit/Paste/Delete/Follow Wire

Not used in this application

Columns Allows the user to enable/disable a number of parameter table columns (figure 6.1.4b).

## 6.5 WATCH/RECIPE EDITOR 


The watch/recipe editor is opened by clicking on the Watch/Recipe tool icon, by selecting ‘Watch/Recipe’ in the ‘Views’ menu or by using the short cut <Ctrl>+<A>. The window is in two parts: the left part containing the watch list; the right-hand part containing one or more data sets, initially empty and unnamed.

The Watch/Recipe window is used:

1. To monitor a list of parameters. This list can contain parameters from many different, and otherwise unrelated parameter lists within the same device. It cannot contain parameters from different devices.
2. To create ‘data sets’ of parameter values which can be selected and downloaded to the device in the sequence defined in the recipe. The same parameter may be used more than once in a recipe.

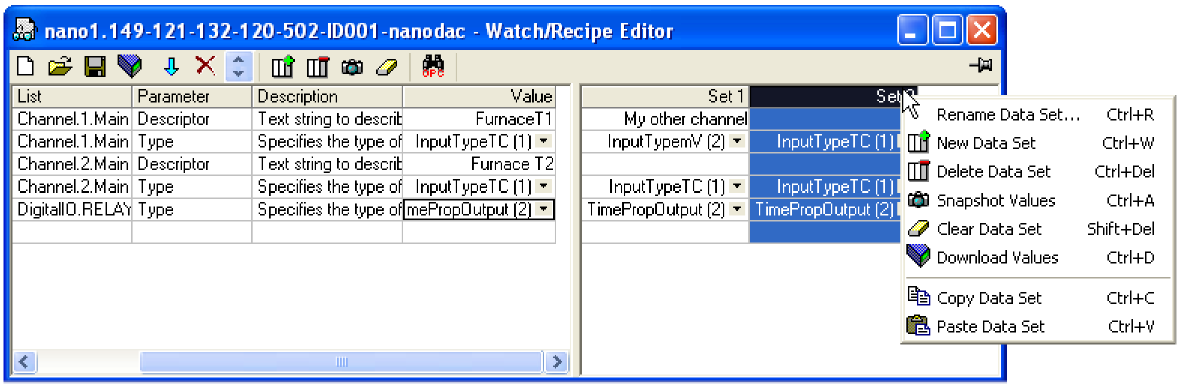



Figure 6.5 Watch/Recipe Editor window (with context menu)

### 6.5.1 Creating a Watch List

After opening the window, parameters can be added to it as described below. The values of the parameters update in real-time, allowing the user to monitor a number of values simultaneously.

#### ADDING PARAMETERS TO THE WATCH LIST

1. Parameters can be click-dragged into the watch list from another area of the iTools window (for example, the parameter explorer window, the graphical wiring editor, the browse tree). The parameter is placed either in an empty row at the bottom of the list, or if it is dragged on top of an already existing parameter, it is inserted above this parameter, with the remaining parameters being moved down one place.
2. Parameters can be dragged from one position in the list to another. In such a case, a copy of the parameter is produced, the source parameter remaining in its original position.
3. Parameters can be copied <Ctrl>+<C> and pasted <Ctrl>+<V> either within the list, or from a source external to it, for example the parameter browse window or the graphical wiring editor.
4. The ‘Insert item...’ tool button the ‘Insert Parameter’ item in the Recipe or context menu or the short cut <Insert> can be used to open a browse window from which a parameter is selected for insertion above the currently selected parameter.

#### DATA SET CREATION

Once all the required parameters have been added to the list, select the empty data set by clicking on the column header. Fill the data set with current values using one of the following methods:

1. Clicking on the ‘Capture current values into a data set’ tool icon (also known as the ‘Snapshot Values’ tool).
2. Selecting ‘Snapshot Values’ from the Recipe or Context (right-click) menu.
3. Using the short cut <Ctrl>+<A>.

### 6.5.1 CREATING A WATCH LIST (Cont.)

**DATA SET CREATION (Cont.)**

Individual data values can now be edited by typing directly into the grid cells. Data values can be left blank or cleared, in which case, no values will be written for those parameters at download. Data values are cleared by deleting all the characters in the cell then either moving to a different cell or typing <Enter>.

The set is called 'Set 1' by default, but it can be renamed by either by using the 'Rename data set...' item in the Recipe or context menus, or by using the short cut <Ctrl>+<R>.

New, empty data sets can be added using one of the following:

1. Clicking on the 'Create a new empty data set' toolbar icon.
2. Selecting 'New Data Set' in the Recipe or context menus
3. Using the short cut <Ctrl>+<W>

Once created, the data sets are edited as described above.

Finally, once all the required data sets have been created, edited and saved, they can be downloaded the instrument, one at a time, using the Download tool, the 'Download Values' item in the Recipe or context menus, or the short cut <Ctrl>+<D>.

### 6.5.2 Watch Recipe toolbar icons

Create a new watch/recipe list. Creates a new list by clearing out all parameters and data sets from an open window. If the current list has not been saved, confirmation is requested. Short cut <ctrl>+<N>

Open an existing watch/recipe file. If the current list or data set has not been saved, confirmation is requested. A file dialogue box then opens allowing the user to select a file to be opened. Short cut <ctrl>+<O>

Save the current watch/recipe list. Allows the current set to be saved to a user specified location. Short cut <ctrl>+<S>.

Download the selected data set to the device. Short cut <ctrl>+<D>

Insert item ahead of selected item. Short cut <Insert>.

Remove recipe parameter. Short cut <ctrl>+<Delete>.

Move selected item. Up arrow moves selected parameter up the list; down arrow move the selected parameter down the list.

Create a new empty data set. Short cut <ctrl>+<w>.

Delete an empty data set. Short cut <ctrl>+<Delete>

Capture current values into a data set. Fills the selected data set with values. Short cut <ctrl>+<A>.

Clear the selected data set. Removes values from the selected data set. Short cut <Shift>+<Delete>.

Open OPC Scope. Opens a separate utility that allows trending, data logging and Dynamic Data Exchange (DDE). OPC Scope is an OPC explorer program that can connect to any OPC server that is in the windows registry.

(OPC is an acronym for 'OLE for Process Control, where OLE stands for 'Object Linking and Embedding'.)

### 6.5.3 Watch/Recipe Context Menu

The Watch/Recipe Context menu items have the same functions as described above for toolbar items.

## 6.6 PROGRAMMER OPTION


Clicking on the Programmer tool bar icon opens the programmer configuration window, displaying the program currently loaded in the instrument, in Segment Parameter view. If no program is loaded, the programmer display opens with just one segment, defined as an 'End' Segment.

Figure 6.6 shows a simple program for example purposes. Parameters are defined in section 3.4.9 and section 4.8.

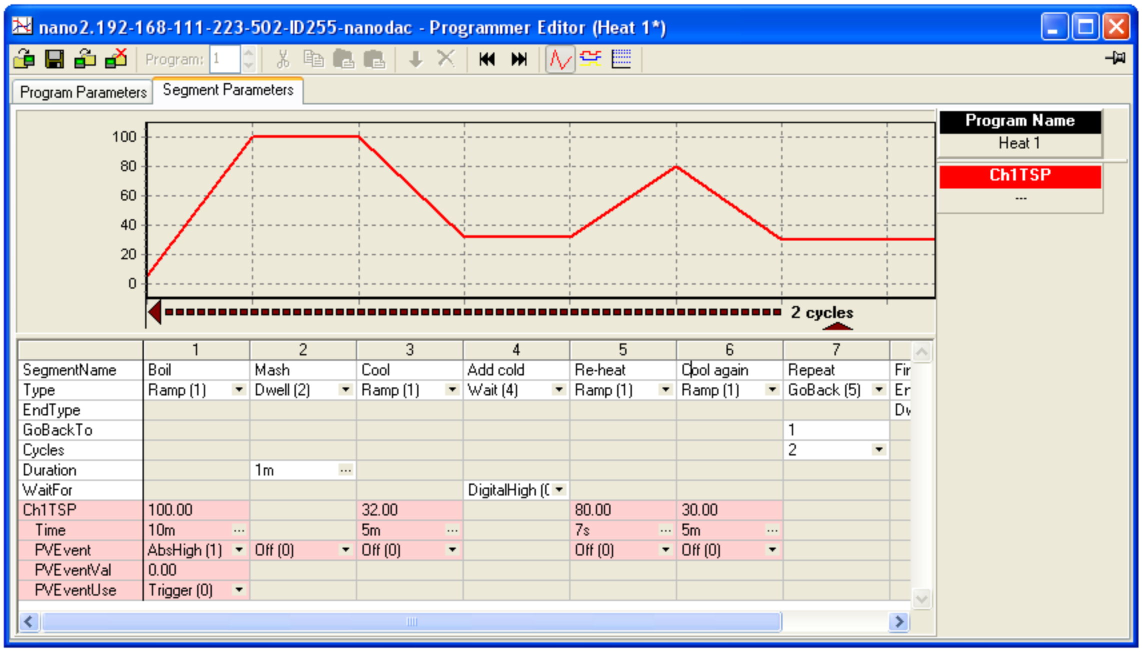
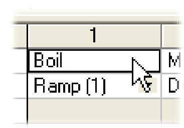



Figure 6.6 Programmer display

As can be seen from the example, the segments appear below a graphical representation of the program.

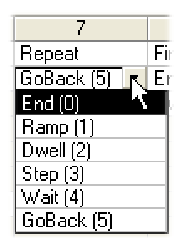
### 6.6.1 Segment parameter editing

#### SEGMENT NAME



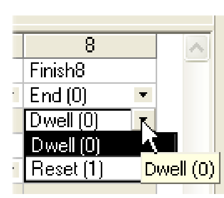

To edit the segment name, click in the segment name field (as shown), and type in the required text, of up to 20 characters. Alternatively, double click on the existing name and edit it as desired.

#### SEGMENT TYPE




Clicking on the down arrow symbol to the right of the existing segment type field, produces a pick list from which a segment type can be selected. The type of segment selected defines which configuration fields appear for that segment.

#### END TYPE




Allows the selection of 'Dwell' or 'Reset' as the action to be taken by the End segment.

### 6.6.1 SEGMENT EDITING (Cont.)

#### GO BACK TO

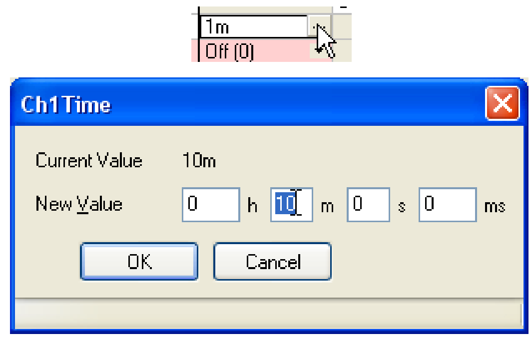
For GoBack segments only, this allows the user to enter a segment number for the program to return to.

#### CYCLES

For GoBack segments only, this allows the number of times the program returns to the 'Go Back To' segment, before continuing.

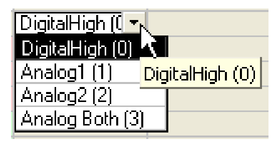
#### DURATION

Sets the amount of time for which Dwell segments are to operate. Times are entered using a hours/minutes/seconds/milliseconds display which appears when the ellipsis button to the right of the duration field is clicked on.




#### WAIT FOR

Select an analogue or digital input as the wait criterion. For single channel programs only one analogue input is available; for two-channel programmers one digital and two analoge inputs are available, as shown.




#### CH1 (2) TSP

The channel 1 (2) target setpoint, editable by the user in a similar way as that used for segment name editing, described above. Ch2 TSP appears only for two channel programmers.

#### TIME

For programs where 'Ramp Style' = 'Time', this allows the user to enter time periods for ramp segments, in a similar way, as described for 'Duration', above. For two channel programmers, two times can be entered, and if the two times are different, the channel with the shorter time waits at its setpoint value until the other channel's time has elapsed.

#### RATE

For programs where 'Ramp Style' = 'Rate', this allows the user to eneter a rate value for Ramp segments. This value is entered in the same way as that used for segment name editing, described above. For two channel programmers, two rates can be entered.

#### OTHER PARAMETERS

Holdback, PV Event etc. parameters may or may not appear depending on the programmer features enabled, and they are all edited in the ways described above.

### 6.6.2 Digital Event display

Clicking on the 'Digital Events Output' tool bar icon produces a segment display, allowing the user to select the events on or off as required, for each segment. Figure 6.6.2 shows a programmer where the number of events is four.

The number of events which appear (maximum eight) is configured in the Programmer Setup menu as described in section 4.8.3

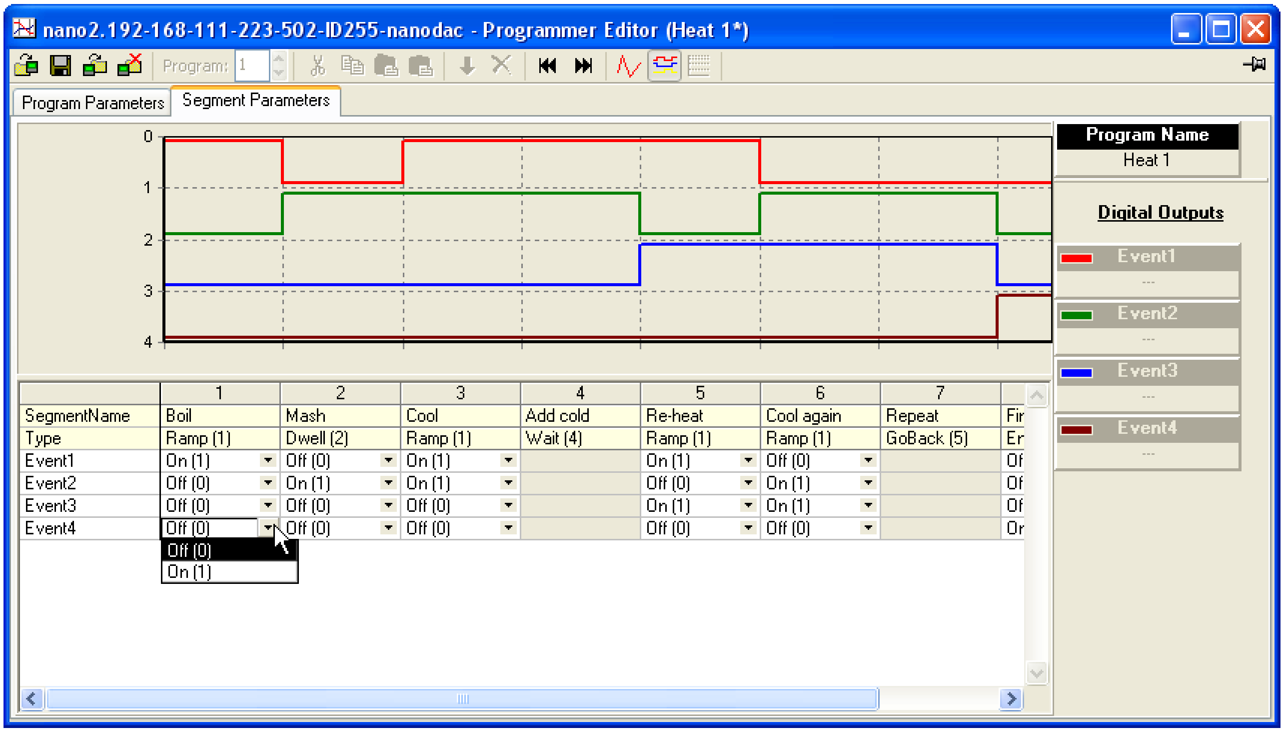



Figure 6.6.2 Event on/off configuration

### 6.6.3 Program parameters

The number of parameters which appear in this display depends on which program features are enabled. Figure 6.6.3 shows a basic set of parameters which allows the user to select Rate or Time as the Ramp style, and to select a value for Rate units.

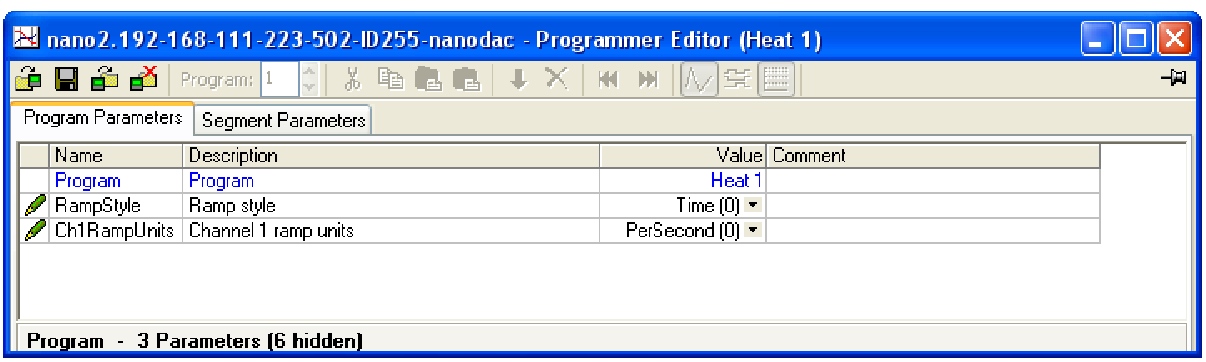



Figure 6.6.3 Program parameter display

### 6.6.4 Adding and deleting segments.

#### INSERT SEGMENT

As shown in figure 6.6.4, to insert a segment, click in the segment number field of the segment to the right of where the new segment is to be located. This causes the whole segment to highlight. Click on the blue down arrow tool icon to insert the new segment. The new segment name is the segment number, and the segment configuration is that of the segment to the right, unless that segment is a dwell or End segment, in which case the new segment is a ramp segment.

To insert more than one segment, operate the shift key whilst clicking on the range of contiguous segments to be copied.

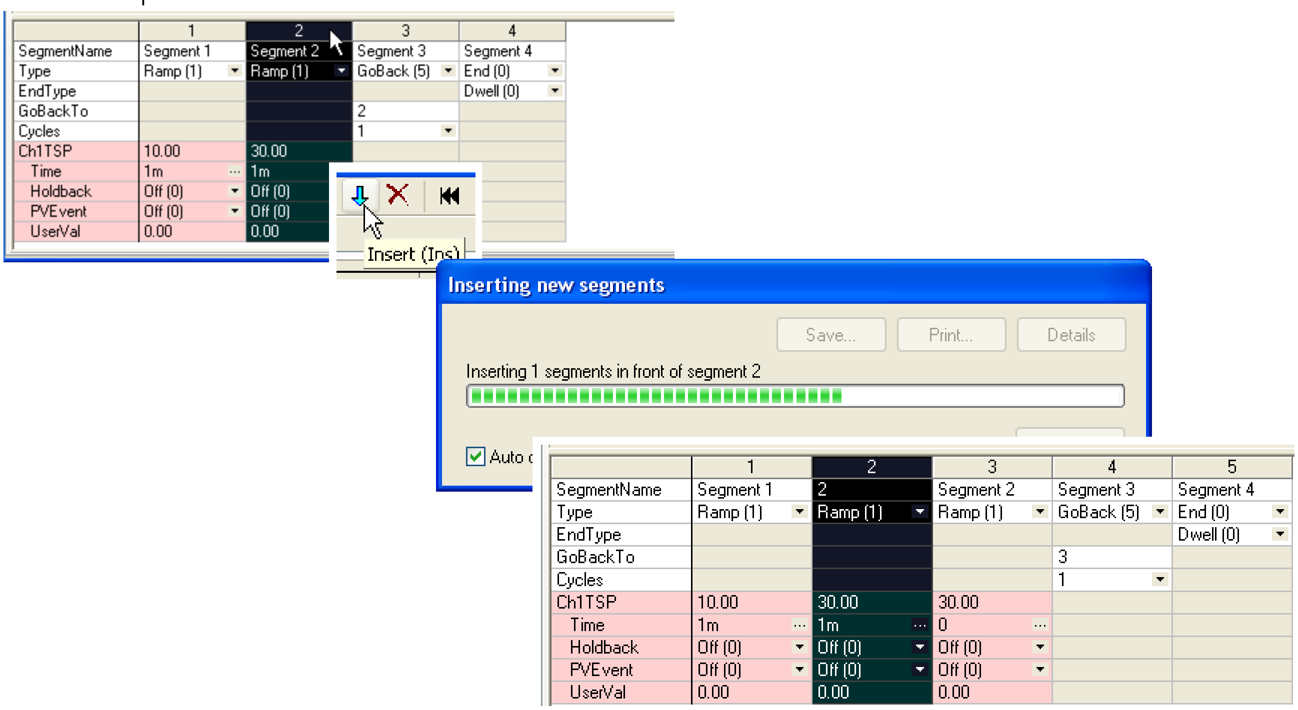
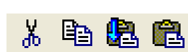



Figure 6.6.4 Insert a segment

Alternatively the mouse can be right-clicked anywhere in a segment, and the 'Insert segment' item selected, or one or more segment can be highlighted and the 'Insert' key on the pc keyboard used to initiate the process. See section 6.6.7 for more details of the right-click (context) menu.

#### CUTTING, COPYING AND PASTING SEGMENTS 

The process of highlighting one or more segments causes the cut and copy toolbar icons to become active.

The cut tool removes the highlighted segments from the program and stores them on the pasteboard ready for re-use.

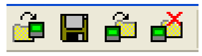
The copy tool copies the selected segment(s) to the paste board, leaving the original segment(s) in place.

Once one or more segments have been cut or copied, the 'Paste insert' and 'Paste over' icons become active allowing the user to paste the contents of the pasteboard in front of a selected segment (Paste insert), or to overwrite the existing highlighted segment(s) (Paste over). When using the Paste over tool, the number of segments being pasted over must match the number of segments on the paste board.

#### DELETING SEGMENTS

Once one or more segments have been highlighted, the highlighted segments can be removed using the Delete toolbar icon, by using the Delete Segment item in the right-click (context) menu, or by operating the pc keyboard 'Ctrl' and Delete' keys simultaneously.

## 6.6.5 Loading and Saving programs

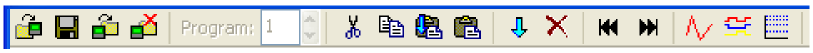


The four program operation keys at the top left of the programmer window allow the user to load a program from or save a program to either the currently connected instrument or to a pc.

The fourth icon allows the user to select a program to be deleted from the connected instrument.

See section 6.6.6 for more details.

## 6.6.6 Toolbar icons




The toolbar icons appearing at the top of the programmer window have the following properties:

- Load Program. Opens a browser window allowing the user to select a program on the pc, or a program stored in the connected instrument to load. Short cut: <Ctrl> + <L>.
- Save current program to file. Opens a browser window allowing the user to select a location on the pc in which to save the current program. This file is saved with a '.upiz' extension and can be saved to a USB memory stick for downloading to an instrument, or it can be transferred to the instrument via an ftp server. Short cut: <Ctrl> + <S>.
- Store current program on device. Allows the user to save the program to the program store on the instrument. Short cut: <Shift key> + <Ctrl> + <S>.
- Delete Programs from Device. Allows the user to delete programs from the program store on the connected. instrument. Short cut: <Ctrl> + <F>.
- Cut. Removes the highlighted segment(s) from the program and places them on the pasteboard. Short cut: <Ctrl> + <X>.
- Copy. Copies the selected segment(s) to the pasteboard, leaving the original segments in place. Short cut: <Ctrl> + <C>.
- Paste insert. Inserts the segments on the pasteboard into a location to the left of the highlighted segment. Short cut: <Ctrl> + <V>.
- Paste over. Overwrites the highlighted segment(s) with the segment(s) on the pasteboard. The number of segments on the pasteboard must match the number of segments being overwritten. Short cut: <Shift key> + <Ctrl> + <V>.
- Insert. Inserts a new segment to the left of the highlighted segment. If more than one segment is highlighted, then the same number of segments are inserted as are highlighted. Copies the segment type of the segment to the right of the insertion point except if that segment is an 'End' or 'GoBack' segment, when newly inserted segments are of type 'Ramp'. Short cut: <Insert>.
- Delete. Deletes the highlighted segment(s). Short cut: <Ctrl> + <Delete>.
- Go to first. Moves the user to the first segment. Useful in very long programs. Short cut: <Ctrl> + <Left arrow>.
- Go to last. Moves the user to the end segment. Useful in very long programs. Short cut: <Ctrl> + <Right arrow>.
- Analog. Selects the analogue trace chart for display and segment configuration. Short cut: <Ctrl> + <G>.
- Digital Event Outputs. Selects the Event output chart for display and configuration.Short cut: <Ctrl> + <D>.
- Logarithmic. Switches the vertical scale to logarithmic. Short cut: <Ctrl> + <M> (figure 6.6.6)

### 6.6.6 TOOLBAR ICONS (Cont.)

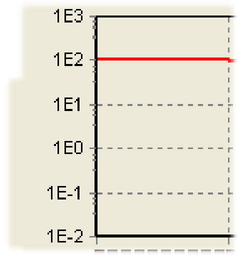
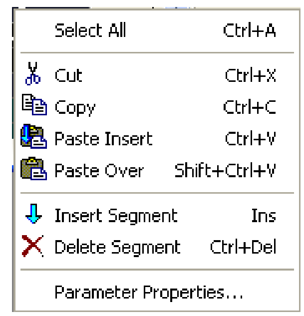



Figure 6.6.6 Logarithmic scale example

## 6.6.7 Context menus

### SEGMENT CONTEXT MENU

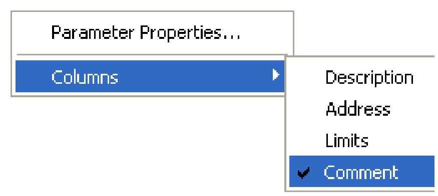
Right-clicking when the mouse cursor is hovering over a segment in the analogue segment parameters view produces the segment context menu shown. The various items copy the relevant tool bar icons described above, with the following additions:




| | |
|---|---|
| Select All | Selects all parameters |
| Parameter properties | Displays the properties window for the parameter right-clicked on, including a 'Help' tag for that parameter. |

### PROGRAM CONTEXT MENU

Right-clicking when the mouse cursor is hovering in the program parameters view produces the program context menu shown.




| | |
|---|---|
| Parameter properties | Displays the properties window for the parameter right-clicked on, including a 'Help' tag for that parameter. |
| Columns | Allows the user to enable/disable columns in the program parameters display. |

### CHART CONTEXT MENU

Right-clicking when the mouse cursor is hovering over the analogue chart or the digital event chart produces the segment context menu shown. This allows the user to copy the chart to the pasteboard, from where it can be pasted into (for example) a standard word procesing document.

### 6.6.8 Programmer menu

Clicking on the 'Programmer' menu item near the top of the iTools window causes the Programmer menu (figure 6.6.8) to appear. The items contained within this menu are described in the 'Toolbar icons' and 'Context menu' sections (sections 6.6.6 and 6.6.7 respectively) above.

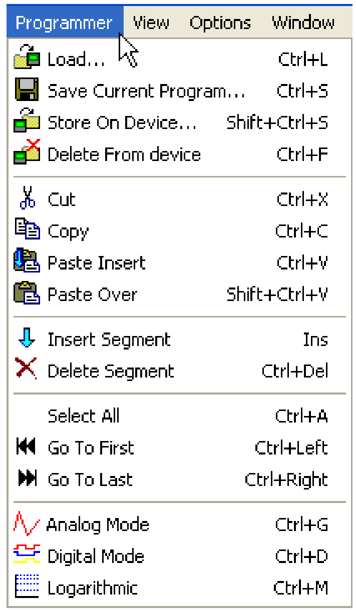



Figure 6.6.8 Programmer menu

### 6.6.9 Two channel programs

The display and editing of segment and program parameters for two-channel programmers is carried out in the same way as described above, for single channel programs. The major difference in apearance is that there are two sets of parameters for each segment, instead of one. The background colour for channel 1 parameters is pink; that for channel 2 parameters is green.

The number of channels and the program features enabled are set up at the instrument as desribed in section 3.4.9 and section 4.8.

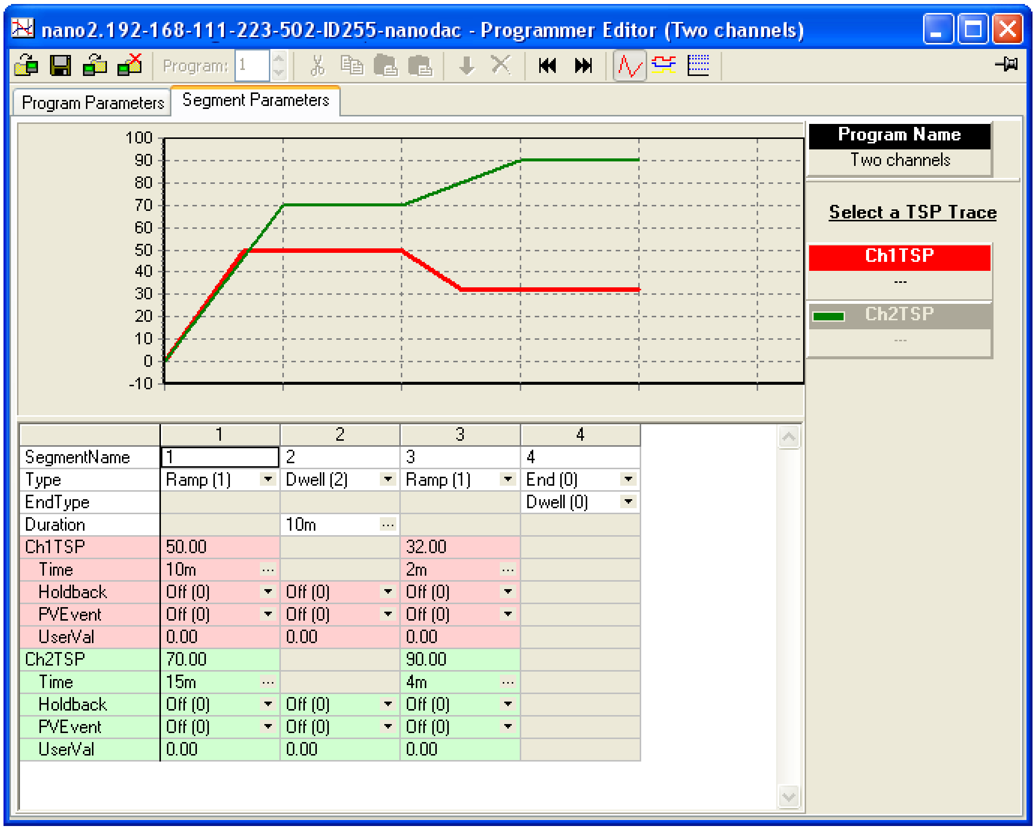



Figure 6.6.9 Two channel program display

# 7 USER WIRING

User wiring, created from the instrument front panel, allows parameters to be wired together so that, for example, a counter can be configured to be incremented when an alarm goes active. This can be used as an alternative to iTools.

This section is presented as two examples that show the general techniques used to create and delete wires from the instrument user interface.

Notes:

1. These examples refer to Channel Configuration and to Virtual Channel configuration, descriptions of which are to be found in sections 4.4 and 4.5 respectively.
2. The destination parameter field has a small green triangle at the top left corner to indicate that it has a wire routed to it. 


## 7.1 DRIVE RELAY EXAMPLE

To drive the relay whose terminal contacts are 3A/3B, whilst the temperature being measured by Channel 2 exceeds 30°C. For this example Channel 2 alarm 1 and a hysteresis of 4°C will be used.

1. In channel 2, Alarm 1 page (see note), set the following parameters:
   Type: Abs. High
   Threshold: 30
   Hysteresis: 4
   Latch: None
   Block: Off
   Dwell:00:00:00
   Acknowledge: No

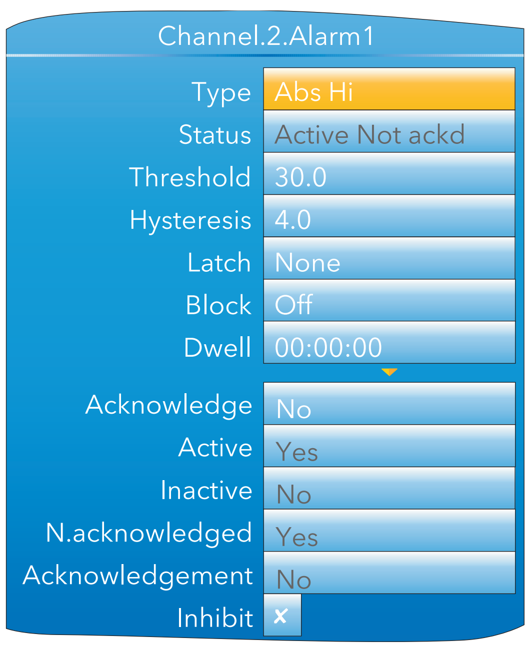
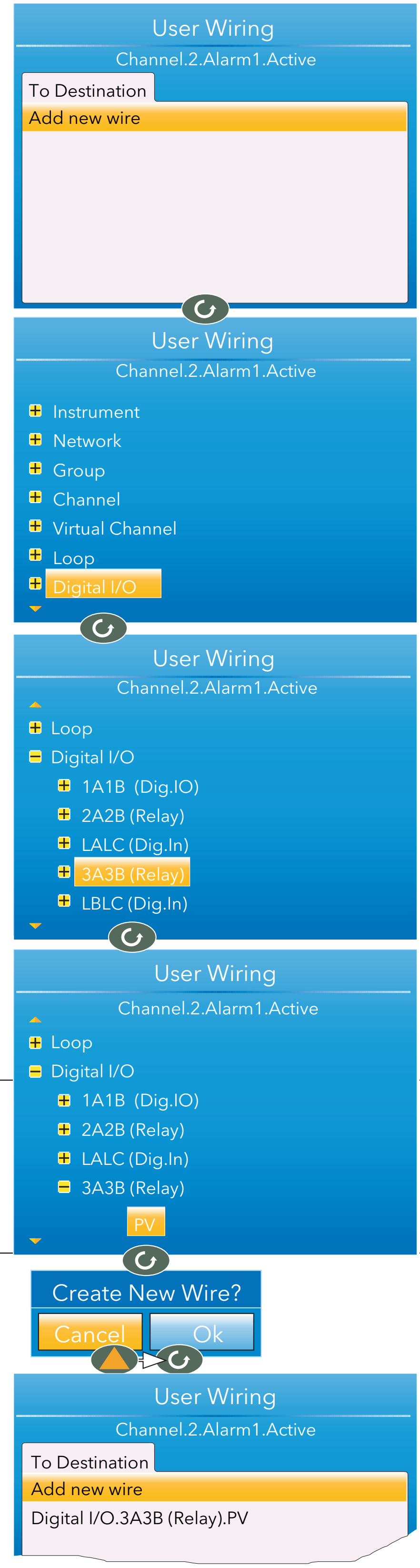



Figure 7.1a Channel 2, Alarm 1 set up

Note: the channel alarm areas of configuration become accessible only once the channel with which they are associated has been configured with a suitable 'Type' (section 4.4.1).

## 7.1 DRIVE RELAY EXAMPLE (Cont.)




2. Highlight the ‘Active’ field, and press and hold the scroll button for a few seconds, until the top level User Wiring page appears.
   The name of the selected parameter appears at the top of the page. Any already existing wires from this parameter would appear below the ‘Add new wire’ area.

3. With ‘Add new wire’ highlighted operate the Scroll button.

4. Use the down arrow to highlight ‘Digital I/O’ and press the scroll button.

5. Use the down arrow to highlight ‘3A3B (Relay)’ and press the scroll button.

6. Use the down arrow to highlight ‘PV’ and press the scroll button.

   Note: If this parameter is already wired-to, the ‘wired’ symbol appears to the left of the parameter.

7. When the confirmation window appears, use the up or down arrow to highlight ‘Ok’, then operate the scroll button again.

8. The top level user wiring page reappears, showing the destination parameter.

### 7.1.1 Wire removal

At the top level user wiring page, use the up and down arrow buttons to highlight the wire to be deleted, and operate the scroll key. In the ‘Delete Wire’ confirmation window, highlight ‘Ok’ and operate the scroll key again. The wire is deleted without further confirmation.

## 7.2 COUNTER EXAMPLE

This example shows how to set up a counter to be incremented each time Channel 1 Alarm 1 becomes active, and reset each time channel 2, alarm 1 is acknowledged. For this example, Virtual Channel 3 will be configured as the counter, with a preset value of 0.

1. At Channel.1.Main, set:

   Type = test
   Test Signal = Sine 4 min.
   Scale Low = 0
   Scale High = 100
2. At Channel.1.Alarm1, set:

   Type = Abs Hi
   Threshold = 50
   Latch = None
3. At Channel.2.Main, set:

   Type = Test
   Test Signal = Sine 40 min.
   Scale Low = 0
   Scale High = 100
4. At Channel.2.Alarm 1, set:

   Type = Abs Hi
   Threshold = 90
   Latch = Manual
5. At Virtual Channel.3.Main, set:

   Type = Counter
   Operation = On
   Input = 1

All the other parameters can be left at their defaults.

6. Still at Virtual Channel 3 (Main), use the up/down arrow buttons to highlight 'Trigger'. Press and hold the scroll key. The top level User Wiring page appears, this time with a 'From Source' tab as well as the 'To Destination' tab of example 1. This is because this parameter is read/write, whereas Alarm Active is read only (i.e. its value may be read but not changed).
7. Use the up (or down) arrow button to select the 'From Source' tab.

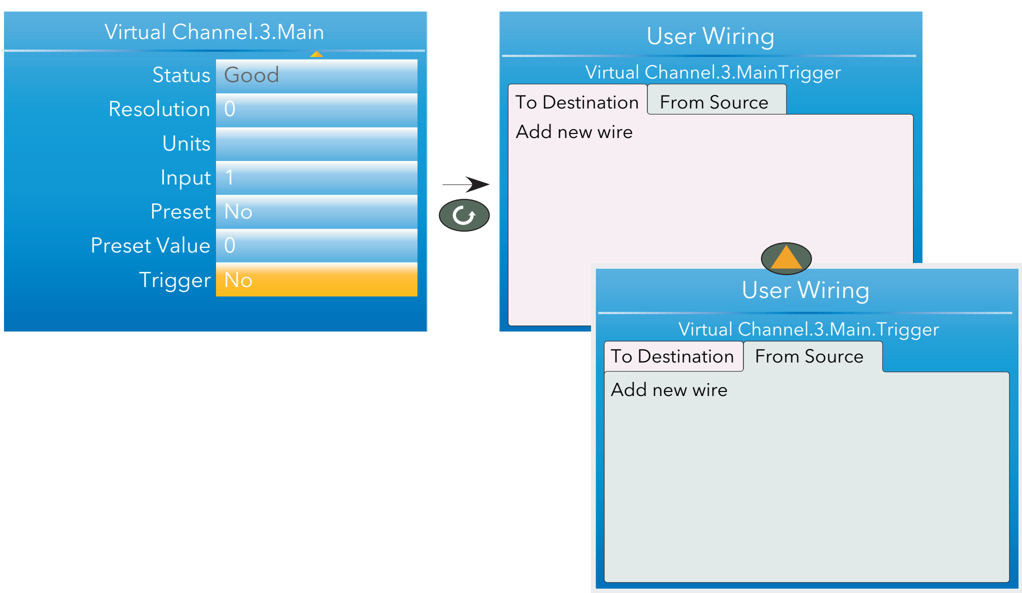



Figure 7.2a Wiring a counter: part 1

## 7.2 COUNTER EXAMPLE (Cont.)

8. Operate the Scroll key to highlight 'Add new wire', then again to display the top level parameter list.
9. Use the down arrow button to highlight 'Channel' and operate the scroll button.
10. Operate the scroll button to select '1'.
11. Highlight 'Alarm 1' and operate the scroll button.
12. Use the down arrow button to highlight 'Active'. Operate the Scroll button again, and create the new wire.
13. Use the Page button twice to return to the Virtual Channel 3 menu.

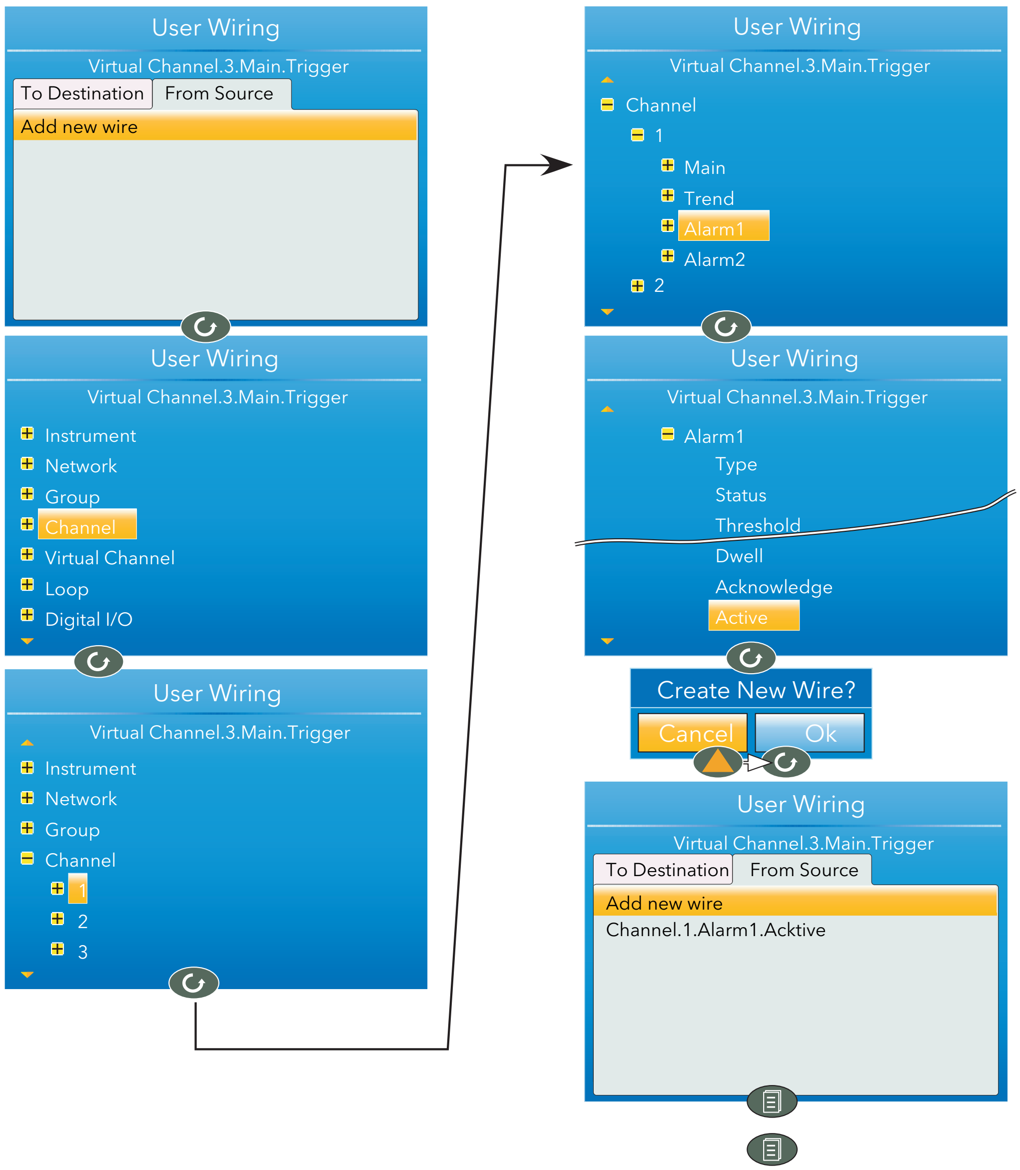



Figure 7.2b Wiring a counter: part 2

## 7.2 COUNTER EXAMPLE (Cont.)

14. At Virtual Channel.3.Main, use the down arrow to select 'Preset'. Press and hold the scroll key. The top level User Wiring page appears.
15. Use the up (or down) arrow button to select the 'From Source' tab, if not already selected.
16. Operate the Scroll key to highlight 'Add new wire', then again to display the top level parameter list.
17. Use the down arrow button to highlight 'Channel' and operate the scroll button.
18. Use the down arrow button to highlight '2' and operate the scroll button.
19. Highlight 'Alarm 1' and operate the scroll button.
20. Use the down arrow button to highlight 'Acknowledgement' (not 'Acknowledge'). Operate the Scroll button again, and create the new wire.

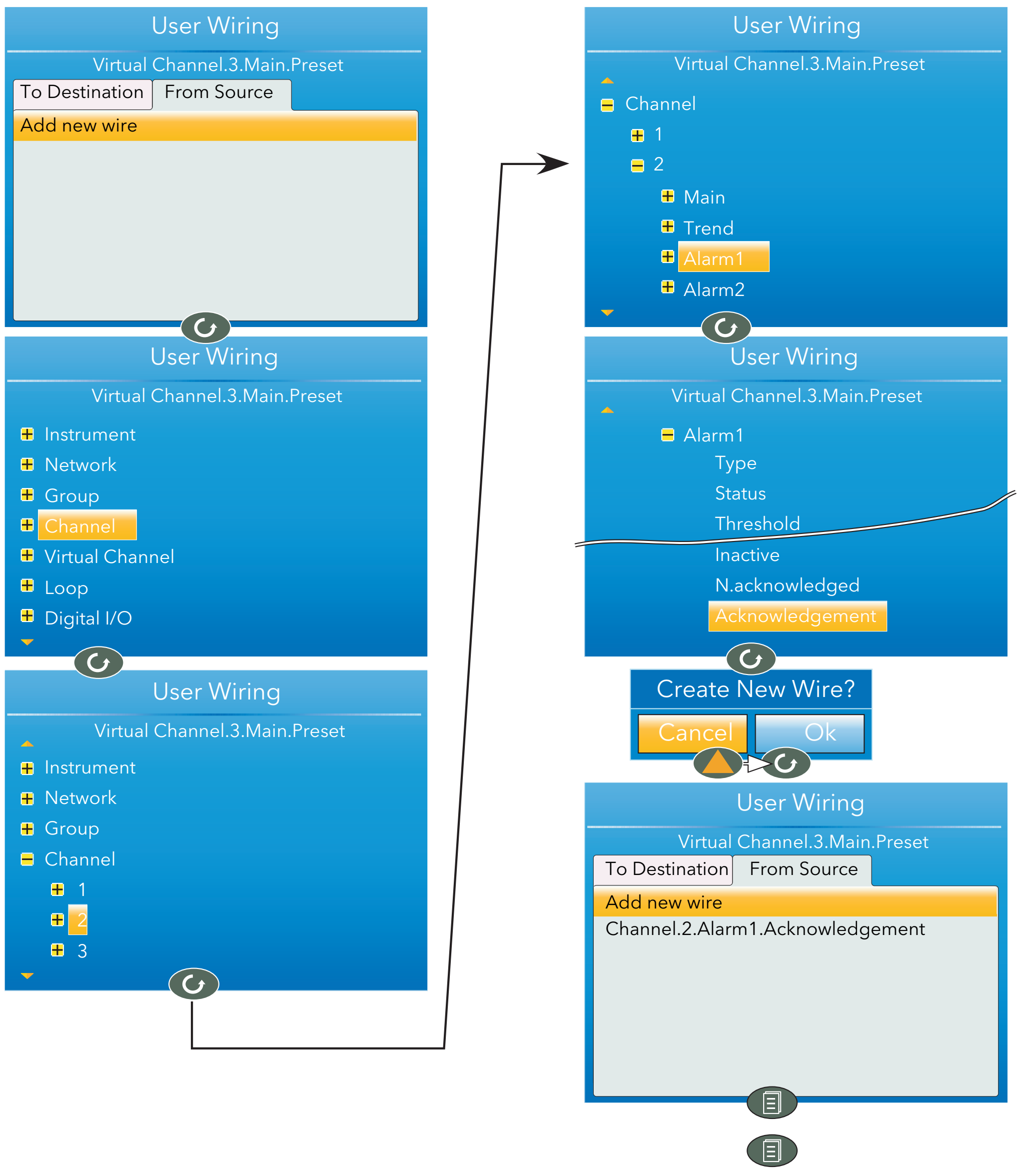



Figure 7.2c Wiring a counter: part 3

# 8 USB DEVICES

The devices listed below can be plugged into the USB connector at the back of the instrument, providing that the maximum current required is less than 100 mA.

1. Memory Stick
2 Bar code reader
3. Keyboard

Notes:
1. See 'USB device precautions' in the Safety Notes preamble section of the manual.
2. See Section A2 for the USB port specification
3. The use of USB hubs is not supported by this instrument.

## 8.1 MEMORY STICK

The use of the memory stick as an archiving device, or to facilitate software upgrades is well documented in the relevant sections of this manual.

## 8.2 BAR CODE READER

If 'USB Auto Scan is set to 'Yes' in Display Configuration (section 4.1.3) then, with the bar code reader plugged into the USB port, the scanned data input stream is packaged into a general message displayed on the trend page and included in the .uhh history file. The format of the message is: DD/MM/YY HH:MM:SS 123--13 (where 123--13 represents the ASCII data read from the bar code.

If 'USB auto Scan' is set to 'No, the ASCII data read from the bar code is displayed as a message ready for editing prior to being sent to the display etc. Figure 8.2 shows an example.

Note: the bar code reader must be configured to use a carriage return (decimal 13) terminating character.

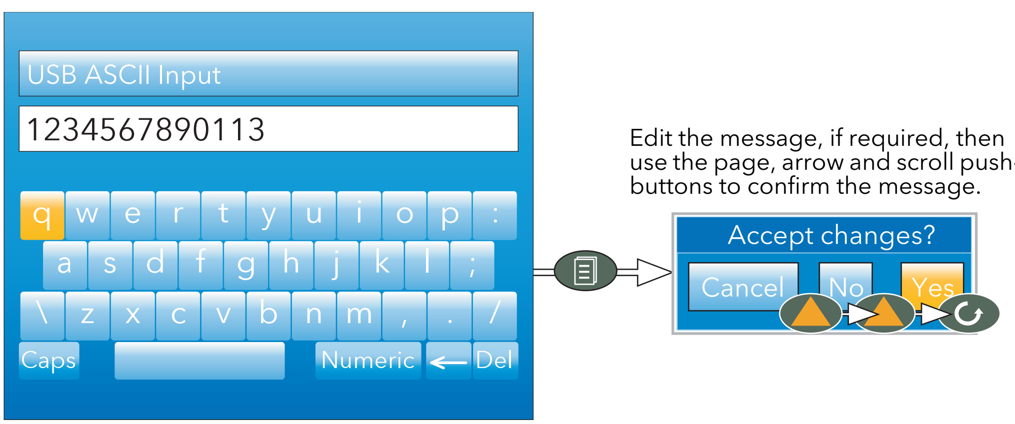



Figure 8.2 Bar Code reader display

## 8.3 USB KEYBOARD

A QWERTY keyboard may be plugged into the rear USB port to act in parallel with the virtual keyboard. The editing keys listed below are supported in addition to the standard alpha-numeric characters.

| | |
|---|---|
| Left arrow | Moves the cursor left-wards through the text string (stops at the start of the string). |
| Right arrow | Moves the cursor right-wards through the text string (stops at the end of the string). |
| Backspace | Deletes the character immediately to the left of the cursor. |
| Delete | Removes the character immediately to the right of the cursor. |
| End | Moves the cursor to the end of the string |
| Home | Moves the cursor to the start of the string |
| Insert | Highlights the entire string, for overwriting |

# Appendix A:  TECHNICAL SPECIFICATION

## A1  INSTALLATION CATEGORY AND POLLUTION DEGREE

This product has been designed to conform to BS EN61010 installation category II and pollution degree 2, defined as follows:

### Installation category II

The rated impulse voltage for equipment on nominal 230V mains is 2500V.

### Pollution degree 2

Normally, only non-conductive pollution occurs.  Occasionally, however, a temporary conductivity caused by condensation shall be expected.

# A2 RECORDER SPECIFICATION

## I/O types

| | |
|---|---|
| Analogue i/p | Four (eight if dual input option fitted) |
| Digital i/p | Two |
| Digital (logic) o/p | See table A2 for options |
| Relay o/p | See table A2 for options |
| DC output | See table A2 for options |
| Features | CSV archive format<br>EtherNet/IP (Option)<br>File transfer protocol (FTP)<br>Messages<br>Modbus Master (Option)<br>Modbus TCP slave<br>Set point programmer (option)<br>uhh (history file) archiving<br>USB port at rear of instrument<br>User linearisation tables (four)<br>Two control loops (optional)<br>Advanced Loop (optional)<br>Zirconia probe support (optional)<br>15 Virtual channels (each configurable as maths, totaliser or counter).<br>30 Virtual channels if Modbus Master or EtherNet/IPoptions fitted (no alarms on virtual channels 16 to 30) |

| IO1 | OP2 | OP3 | OP4 | OP5 | |
|---|---|---|---|---|---|
| L | R | R | R | R | Default |
| L | R | D | R | R | Options |
| L | L | R | R | R | |
| R | D | D | R | R | |
| D | D | D | R | R | |

L = Logic output; R = Relay; D = DC output
OP4 and OP5 share Common terminals.

Table A2 Output options

## Environmental performance

| | |
|---|---|
| Ambient temperature range | |
| Operating: | 0 to 55°C |
| Storage: | -20 to +70°C |
| Humidity range Operating: | 5% to 85% RH non condensing |
| storage: | 5% to 85% RH non condensing |
| Protection Front panel (Standard): | IP65 |
| Front panel (Wash down): | IP66, NEMA4X (International) |
| Behind panel: | IP10 (International) |
| Shock/Vibration | To BS EN61131-2 (5 to 150 Hz. at 1g; 1 octave per min.) |
| Altitude | <2000 metres |
| Atmosphere | Not suitable for use in explosive or corrosive atmospheres. |
| Electrical safety | BS EN61010-1 (Installation category II; Pollution degree 2) |
| Electromagnetic compatibility | |
| Emissions (standard units): | BS EN61326 Class B - Light industrial. |
| Emissions (Low voltage option): | BS EN61326 Class A - Heavy industrial |
| Immunity | BS EN61326 Industrial |

## Other approvals and compliance details

| | |
|---|---|
| General: | CE and cUL, EN61010 |
| PV input | AMS2750D compliant |
| RoHS | EU; China |
| Packaging | BS EN61131-2 section 2.1.3.3. |

## Physical

| | |
|---|---|
| Panel mounting | 1/4 DIN |
| Weight | 0.44kg (15.52 oz.) |
| Panel cutout dimension | 92 mm x 92 mm (both -0.0 +0.8) or 3.62 in x 3.62 in (both -0.00 +0.03 in) (figure 2.1) |
| Depth behind panel | 90 mm (3.54 in) (figure 2.1) excluding wiring. |

## Operator interface

| | |
|---|---|
| Display | 3.5" TFT colour display (320 pixels wide x 240 pixels high) |
| Controls | Four navigation pushbuttons below the display screen (Page, Scroll, Lower and Raise) |

## Power requirements

| | |
|---|---|
| Supply voltage Standard: | 100 to 230Vac ± 15% at 48 to 62Hz. |
| Low voltage option: | 24Vac (+10% - 15%), at 48 to 62 Hz, or 24Vdc (+20% -15%) |
| Power dissipation | 9 W |
| Fuse type | None |
| Interrupt protection Standard: | Holdup >10ms at 85V RMS supply voltage. |
| Low voltage option: | Holdup >10ms at 20.4V RMS supply voltage. |

## Battery backup

| | |
|---|---|
| Stored data | Time, date. |
| Support time (for real-time clock) | Minimum of 1 year with unit unpowered. |
| Replacement period | Three years typical |
| Type | poly-carbonmonofluoride / lithium (BR2330) (PA260195) |

## Ethernet communications

| | |
|---|---|
| Type: | 10/100baseT Ethernet (IEEE802.3) |
| Protocols: | Modbus TCP/IP slave, FTP, DHCP |
| Cable type | Category 5 |
| Maximum length | 100metres (110 yards) |
| Termination | RJ45. Green LED illuminated = link connected; Amber LED flashing shows link activity. |

## A2 RECORDER SPECIFICATION (Cont.)

USB port

| | |
|---|---|
| Number of ports | One at rear of instrument |
| Standard | USB1.1 |
| Transmission speeds | 1.5Mbits/sec (low speed device) |
| Maximum current | <100mA |

Peripherals supported Memory stick (8GB max), Bar code reader, QWERTY keyboard

Update/Archive rates

| | |
|---|---|
| Sample rate (input/output) | 8Hz (4Hz for digital inputs) (4Hz for dual input channels) |
| Trend update | 8 Hz max |
| Archive sample value | Latest value at archive time |
| Display value | Latest value at display update time. |

# A3 ANALOGUE INPUT SPECIFICATION

General

| | |
|---|---|
| Number of anaologue inputs | Four |
| Input types Standard: | dc Volts, dc mV, dc mA (external shunt required), thermocouple, RTD (2-wire and 3-wire), digital (contact closure). |
| Optional: | dual mA, dual mV, dual thermocouple. |
| Input type mix | Freely configurable |
| Sample rate | 8Hz (125ms) |
| Conversion method | 16 bit delta sigma. |
| Input ranges | See below. |
| Mains rejection (48 to 62Hz) | |
| Series mode: | >95dB |
| Common mode: | >179dB |
| Common mode voltage | 250Vac max. |
| Series mode voltage | 280mV at lowest range; 5V peak-to-peak, at highest range. |
| Input impedance | See relevant Range specification, below. |
| Overvoltage protection Continuous: | ± 30V RMS |
| Transient (<1ms): | ±200V pk-pk between terminals. |
| Sensor break detection Type | ac sensor break on each input giving quick response with no associated dc errors. |
| Recognition time: | <3 secs. |
| Minimum break resistance: | 40mV, 80mV ranges: 5kΩ; other ranges: 12.5kΩ |
| Shunt (mA inputs only) Values | 1Ω to 1kΩ, mounted externally. |
| additional error due to shunt: | 0.1% input |
| Isolation Channel to channel: | 300V RMS or dc (double insulation) |
| Channel to common electronics: | 300V RMS or dc (double insulation) |
| Channel to ground: | 300V RMS or dc (double insulation) |
| Dielectric strength Test: | BS EN61010, 1 minute type test |
| Channel to channel: | 2500 Vac |
| Channel to ground: | 1500 Vac |

DC input ranges

| | |
|---|---|
| Ranges | 40mv, 80mV, 2V; 10V (-4.0 to +10V) |
| 40mV Range Range: | -40mV to + 40mV |
| Resolution | 1.9µV (unfiltered) |
| Measurement noise: | 1.0µV peak-to-peak with 1.6s input filter |
| Linearity error: | 0.003% (best fit straight line) |
| Calibration error: | ±4.6µV ±0.053% of measurement at 25°C ambient |
| Temperature coefficient: | ±0.2µV/°C ± 13ppm/°C of measurement from 25°C ambient |
| Input leakage current: | ±14nA |
| Input resistance: | 100MΩ |
| 80mV Range Range: | -80mV to + 80mV |
| Resolution | 3.2µV (unfiltered) |
| Measurement noise: | 3.3µV peak-to-peak with 1.6s input filter |
| Linearity error: | 0.003% (best fit straight line) |
| Calibration error: | ±7.5µV ±0.052% of measurement at 25°C ambient |
| Temperature coefficient: | ±0.2µV/°C ± 13ppm/°C of measurement from 25°C ambient |
| Input leakage current: | ±14nA |
| Input resistance: | 100MΩ |
| 2V Range Range: | ±2V |
| Resolution | 82µV |
| Measurement noise: | 90µV peak-to-peak with 1.6s input filter |
| Linearity error: | 0.003% (best fit straight line) |
| Calibration error: | ±420µV ±0.044% of measurement at 25°C ambient |
| Temperature coefficient: | ±125µV/°C ± 13ppm/°C of measurement from 25°C ambient |
| Input leakage current: | ±14nA |
| Input resistance: | 100MΩ |

## A3 ANALOGUE INPUT SPECIFICATION (Cont.)

### DC Input ranges (Cont.)

10V Range
Range: -3V to +10V
Resolution 500µV
Measurement noise: 550µV peak-to-peak with 1.6s input filter
Linearity error: 0.007% (best fit straight line) for zero source resistance. Add 0.003% for each 10Ω source and lead resistance
Calibration error: ±1.5mV ±0.063% measurement at 25°C ambient
Temperature coefficient: ±66µV/°C ± 45ppm/°C of measurement from 25°C ambient
Input resistance: 62.5kΩ for input voltages > 5.6V. 667kΩ for input ranges < 5.6V.

Note: 10V range not available for dual input channels

Resistance input ranges
Temperature scale: ITS90
RTD Types, ranges and accuracies: See table
Maximum source current: 200µA

Resistance input figures
Range: 0 to 400Ω (-200 to +850°C)
Resolution: 0.05°C
Measurement noise: 0.05°C peak-peak with 1.6s input filter
Linearity error: 0.0033% (best fit straight line)
Calibration error: ±0.31°C ±0.023% of measurement in °C at 25°C ambient
Temperature coefficient: ±0.01°C/°C ±25ppm/°C measurement in °C from 25°C ambient
Lead resistance: 0 to 22Ω matched lead resistances
Bulb current: 200µA nominal

| RTD type | Overall range °C | Standard | Max. linearisation error |
|---|---|---|---|
| Cu10 | -20 to + 400 | General electric Co. | 0.02°C |
| Cu53 | -70 to + 200 | RC21-4-1966 | <0.01°C |
| JPT100 | -220 to + 630 | JIS C1604:1989 | 0.01°C |
| Ni100 | -60 to + 250 | DIN43760:1987 | 0.01°C |
| Ni120 | -50 to + 170 | DIN43760:1987 | 0.01°C |
| Pt100 | -200 to + 850 | IEC751 | 0.01°C |
| Pt100A | -200 to + 600 | Eurotherm Recorders SA | 0.09°C |

Table A3a RTD type details

Thermocouple data
Temperature scale: ITS90
CJC Types: Off, internal, external, remote.
Remote CJC source: Any input channel
Internal CJC error: <1 °C max, with instrument at 25 °C
Internal CJC rejection ratio: 40:1 from 25°C
Upscale/downscale drive: High, low or none independently configurable for each channel's sensor break detection.
Types, ranges and accuracies: See table A3b

| T/C type | Overall range (°C) | Standard | Max. linearisation error |
|---|---|---|---|
| B | 0 to + 1820 | IEC584.1 | 0 to 400°C = 1.7°C<br>400 to 1820°C = 0.03°C |
| C | 0 to + 2300 | Hoskins | 0.12°C |
| D | 0 to + 2495 | Hoskins | 0.08°C |
| E | -270 to + 1000 | IEC584.1 | 0.03°C |
| G2 | 0 to + 2315 | Hoskins | 0.07°C |
| J | -210 to + 1200 | IEC584.1 | 0.02°C |
| K | -270 to + 1372 | IEC584.1 | 0.04°C |
| L | -200 to + 900 | DIN43710:1985 (to IPTS68) | 0.02°C |
| N | -270 to + 1300 | IEC584.1 | 0.04°C |
| R | -50 to + 1768 | IEC584.1 | 0.04°C |
| S | -50 to + 1768 | IEC584.1 | 0.04°C |
| T | -270 to + 400 | IEC584.1 | 0.02°C |
| U | -200 to + 600 | DIN43710:1985 | 0.08°C |
| NiMo/NiCo | -50 to + 1410 | ASTM E1751-95 | 0.06°C |
| Platinel | 0 to + 1370 | Engelhard | 0.02°C |
| Mi/NiMo | 0 to + 1406 | Ipsen | 0.14°C |
| Pt20%Rh/Pt40%/Rh | 0 to + 1888 | ASTM E1751-95 | 0.07°C |

Table A3b Thermocouple types, ranges and accuracies

# A4 RELAY AND LOGIC I/O SPECIFICATION

OP1, OP2, OP3 logic input, logic output and relay specification.

Active (current on) current sourcing logic output

| | |
|---|---|
| Voltage output across terminals | +11V min; +13V max. |
| Short circuit output current | 6mA min. (steady state); 44mA max. (switch current) |

Inactive (current off) current sourcing logic output (OP1 or OP2 only)

| | |
|---|---|
| Voltage output across terminals | 0V (min.); 300mV (max) |
| Output source leakage current into short circuit | 0µA (min.); 100µA max |

Active (current on) contact closure sourcing logic input (OP1 only)

| | | |
|---|---|---|
| Input current | Input at 12V: | 0mA (min.); 44mA (max.) |
| | inout at 0V: | 6mA min. (steady state); 44mA max. (switch current) |
| Open circuit input voltage | | 11V (min.); 13V (max.) |
| Open circuit (inactve) resistance | | 500Ω (min.); ∞ (max.) |
| Closed circuit (active) resistance | | 0Ω (min.); 150Ω (max.) |

Relay contacts

| | |
|---|---|
| Contact switching power (resistive) | Max: 2A at 230V RMS ±15%; Min: 100mA @ 12V. |
| Maximum current through terminals | 2A |

# A5 DIGITAL INPUTS

DigInA, DigInB, contact closure logic input

Contact closure

| | |
|---|---|
| Short circuit sensing current (source) | 5.5mA (min.); 6.5mA (max.) |
| Open circuit (inactive) resistance | 600 Ω (min.); ∞ (max.) |
| Closed circuit (active) resistance | 0Ω (min.); 300Ω (max.) |
| Maximum frequency | 8 Hz |
| Minimum pulse width | 62.5 ms |

# A6 DC OUTPUTS

OP1, OP2, OP3 DC analogue outputs

Current outputs (OP1, OP2 and OP3)

| | |
|---|---|
| Output ranges | Configurable within 0 to 20mA |
| Load resistance | 500Ω Max. |
| Calibration accuracy | <±100µA ±1% of reading |

Voltage outputs (OP3 only)

| | |
|---|---|
| Output range | Configurable within 0 to 10V |
| Output impedance | 500Ω Min. |
| Calibration accuracy | <±50mV ±1% of reading |

General

| | |
|---|---|
| Isolation | 300Vac double insulated from instrument and other I/O |
| Resolution | >11 bits |
| Thermal drift | <100ppm/°C |

# A7 BLOCKS SUPPORTED

## A7.1 'TOOLKIT' BLOCKS

BCD input
Eight-input logic
Eight input multiplexer
Timers
Two-input logic
Two-input maths
User values

## A7.2 APPLICATION BLOCKS

Humidity
Steriliser
Zirconia

# Appendix B CONTROL LOOPS

Note: See section 4.6 for Loop configuration details

## B.1 INTRODUCTION

With this recorder, two control loops are available, each loop containing two outputs (Channel 1 and Channel 2) which can be individually configured for PID, On/Off or valve position. For temperature control, channel 1 is normally configured for heating and channel 2 for cooling.

### B1.1 EXAMPLE (HEAT ONLY)

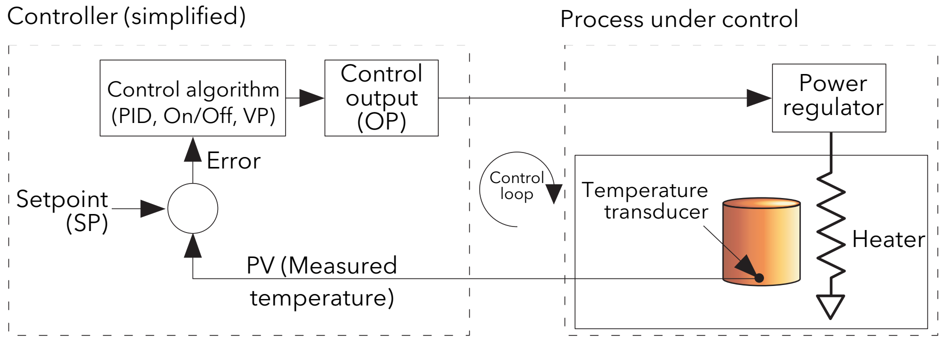



Figure B1.1 Control loop example

The measured temperature (process variable, or 'PV') is connected to the input of the controller, where it is compared with the 'Setpoint' (SP) (the target temperature). If there is a difference between the PV and the SP, the controller calculates and outputs a heating demand. This output is applied to the process heating device, which in turn causes a change in the PV in a direction intended to result in a zero error.

## B2 CONTROL LOOP DEFINITIONS

### B2.1 AUTO/MANUAL

In manual mode, if 'On/Off' control is configured, the output power may be edited by the user but the only power values allowed are: +100% (heat on; cool off) for positive user entries, 0% (heat off; cool off) for zero entry or -100%. (heat off; cool on) for negative entries.

In manual mode, for 'PID' control the output may be edited between +100% and (if 'cool' is configured), -100%. The actual output value is subject to limiting and output rate limit.

In manual mode, for valve position control, the up and down arrow buttons directly control (nudge) the raise and lower relay outputs respectively. It is also possible to control the valve by sending nudge commands over a serial link, or by software wiring from a suitable parameter. A single nudge command moves the valve by 1 minimum on time; longer nudge demands produce longer valve movements. See section B2.6.10 for more details.

If sensor break occurs while the controller is in automatic the controller outputs the sensor break output power. In such a case the user can switch to manual control and edit the output power. On returning to automatic control, the controller checks again for sensor break.

If autotune is enabled while in manual mode, the autotune remains in a reset state such that when the user puts the controller into automatic control the autotune starts.

## B2.2 TYPES OF CONTROL LOOP

### B2.2.1 On/Off control

This form of control turns heating power on when the process value is below the setpoint, and turns it off when it is above the setpoint (see also figure B2.6.9a). If cooling is configured, it has its own relay which operates in a similar way. In Direct Acting mode, the behaviour is inverted. On/off is suitable for controlling switching devices such as relays.

Because of the thermal inertia of the load, a certain amount of oscillation will take place, and this can affect the quality of the product. For this reason, On/Off control is not recommended for critical applications.

Depending on the nature of the process being controlled, some hysteresis may have to be included to prevent continuous operation or chatter in the controlling device.

### B2.2.2 PID Control

Also known as 'three term control', this type of control continuously adjusts the output demand, according to a set of rules, in order to control the process as closely as possible to requirements. PID provides more stable control than On/Off control but is more complex to set up as the parameters must match the characteristics of the process under control.

The three major parameters are: Proportional band (PB), Integral time (Ti) and Derivative time (Td) and the output from the controller is the sum of these three terms. This output is a function of the size and duration of the error value and the rate-of-change of the process value.

It is possible to disable the integral and/or derivative terms and control on proportional only, on proportional plus integral (PI) or proportional plus derivative (PD).

PI control is often used when the PV is noisy and/or subject to rapid variations, where derivative action would cause the output power to fluctuate wildly.

#### PROPORTIONAL BAND

The proportional band (PB) delivers an output which is proportional to the size of the error signal. It is the range over which the output power is continuously adjustable in a linear fashion from 0% to 100% (for a heat only controller). Below the proportional band the output is full on (100%), above the proportional band the output is full off (0%) as shown in figure B2.2.2a.

The width of the proportional band determines the magnitude of the response to the error. If PB is too narrow (high gain) the system oscillates; if it is too wide (low gain) control is sluggish. The ideal situation is when the proportional band is as narrow as possible without causing oscillation.

Figure B2.2.2a also shows the effect of narrowing proportional band to the point of oscillation. A wide proportional band results in straight line control but with an appreciable initial error between setpoint and actual temperature. As the band is narrowed the temperature gets closer to setpoint until eventually, it becomes unstable.

The proportional band may be set in engineering units or as a percentage of the controller range.

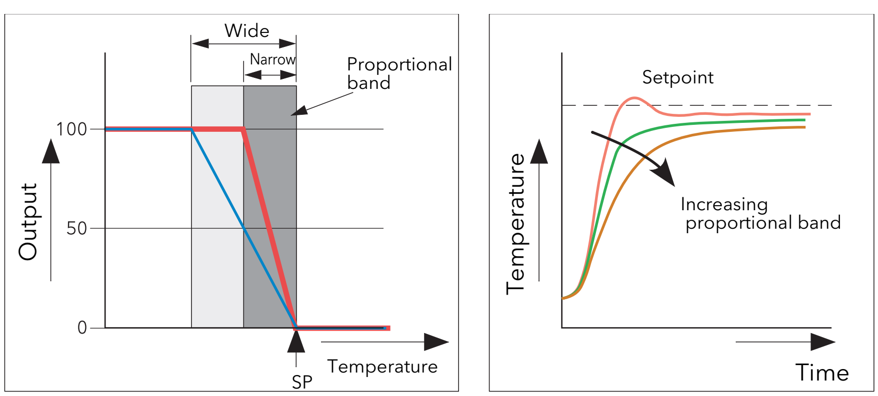



Figure B2.2.2a Proportional band action (reverse acting)

## B2.2 TYPES OF CONTROL LOOP (Cont.)

### INTEGRAL TERM

In a proportional only controller, as seen in the previous section, an error must exist between setpoint and PV in order for the controller to deliver power. Integral is used to achieve zero steady state control error.

The integral term slowly modifies the output level as a result of any error between setpoint and measured value. If the measured value is below setpoint the integral action gradually increases the output in an attempt to correct the error. If it is above setpoint integral action gradually decreases the output or increases the cooling power to correct the error.

Figure B2.2.2b shows proportional plus integral action.

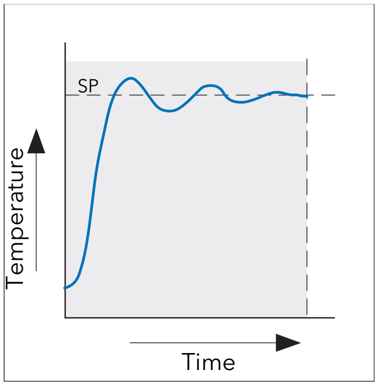



Figure B2.2.2b: Proportional + Integral Control

The integral term is set in seconds. The longer the integral time constant, the more slowly the output is modified and the more sluggish the response. Too small an integral time causes the process to overshoot, and perhaps to start oscillating. The integral action may be disabled by setting its value to Off.

### DERIVATIVE TERM

Derivative (or rate) action provides a sudden change in output linked to the rate of change in error, whether this is caused by PV alone (derivative on PV) or by a change in the SP as well (derivative on error selection). If the measured value falls quickly, derivative provides a large change in output in an attempt to correct the perturbation before it goes too far. It is most beneficial in recovering from small perturbations.

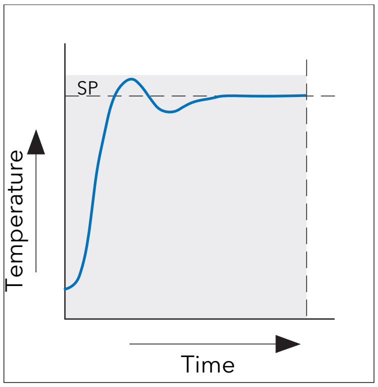



Figure B2.2.2c Proportional + Integral + Derivative Action

Derivative is used to improve the performance of the loop. There are, however, situations where derivative may be the cause of instability. For example, if the PV is noisy, then derivative can amplify that noise and cause excessive output changes, in these situations it is often better to disable the derivative and re-tune the loop.

## B2.2 TYPES OF CONTROL LOOP (Cont.)

Derivative should not be used to curb overshoot in situations when the output is saturated at Op High or Op Low for extended periods, such as process start up, since to do so degrades the steady state performance of the system. Overshoot inhibition is best left to the approach control parameters, High and Low Cutback.

If Derivative is set to Off, no derivative action will be applied.

Derivative can be calculated on change of PV or change of Error. If configured on error, then changes in the setpoint will be transmitted to the output. For applications such as furnace temperature control, it is common practice to select Derivative on PV to prevent thermal shock caused by a sudden change of output as a result of a change in setpoint.

### B2.2.3 Motorised valve control

Designed specifically for driving motorised valves this type of control can operate in 'Unbounded' mode (VPU) or 'Bounded' mode (VPB). Relay outputs are used to drive the valve motor.

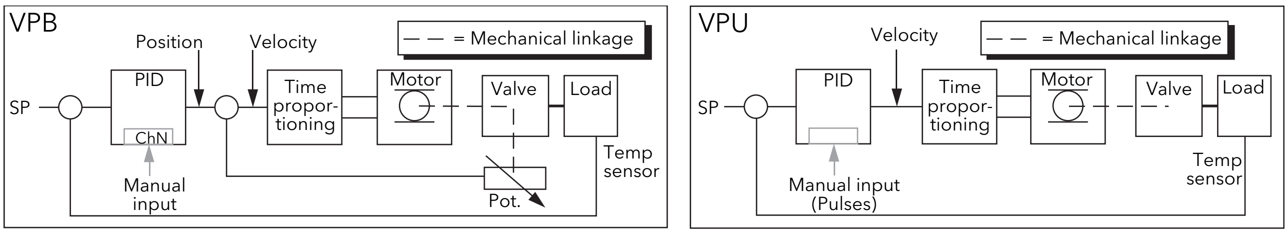



Figure B2.2.3 VPB and VPU comparison

Unbounded valve positioning (VPU) does not require a position feedback potentiometer in order to operate because it controls directly the direction and velocity of the movement of the valve in order to minimise the error between the setpoint (SP) and the process variable (PV). Control is performed by delivering a 'raise' or 'lower' pulse to control the velocity of the valve in response to the control demand signal.

Bounded VP (VPB) control uses PID (or any other combination of the three terms) to set a required valve position. A feedback potentiometer linked to the valve provides a signal giving actual valve position. This allows the control loop to calculate the difference between required and actual position dynamically, and adjust control output accordingly. Control is performed by delivering a 'raise' or 'lower' pulse to adjust the valve position.

### MANUAL MODE

Bounded VP controls in manual mode because the inner positional loop is still running against the potentiometer feedback, so it is operating as a position loop.

In boundless mode the algorithm is a velocity mode positioner. When manual is selected then the up and down arrow produce +100% or –100% velocity respectively for the duration of the key press.

In boundless mode it is essential that the motor travel time is set accurately in order to allow the integral time to calculate correctly. Motor travel time is defined as (valve fully open – valve fully closed). This is not necessarily the time printed on the motor since, if mechanical stops have been set on the motor, the travel time of the valve may be different.

Every time the valve is driven to its end stops the algorithm is reset to 0% or 100% to compensate for any changes which may occur due to wear in linkages or other mechanical parts.

This technique makes boundless VP look like a positional loop in manual even though it is not. This enables combinations of heating and cooling e.g. PID heat, VPU cool with manual mode working as expected.

### MOTORISED VALVE OUTPUT CONNECTIONS

The loop output which has been configured as valve position can be wired to the PV input of one of the pairs of relays 2A2B/3A3B or 4AC/5AC which has been configured as Type = 'Valve Raise'. Only one relay input needs to be wired as the other relay of the pair will be automatically set to 'Valve Lower'. For example, if Loop 1 Channel 1 output is wired to Relay 2A2B and the 'Type' is configured as 'Valve Raise' then the Type for Relay 3A3B will be 'Valve Lower'.

## B2.3 LOOP PARAMETERS

### B2.3.1 Relative cool gain (R2G)

This is the gain of channel 2 control output, relative to the channel 1 control output and is used to compensate for the different quantities of power available to heat and to cool a process. For example, water cooling applications might require a relative cool gain of 0.25 because cooling is 4 times greater than the heating process at the operating temperature.

By default, this parameter is set automatically when an Autotune is performed, but setting the Tune menu parameter 'AT.R2G' to 'No' causes the R2G value(s) entered in the PID menu to be used instead.

### B2.3.2 High and Low cutback

Cutback high 'CBH' and Cutback low 'CBL' are values that modify the amount of overshoot, or undershoot, that occurs during large step changes in PV under start-up conditions, for example. They are independent of the PID terms which means that the PID terms can be set for optimal steady state response and the cutback parameters used to modify any overshoot which may be present.

Cutback involves moving the proportional band towards the cutback point nearest the measured value whenever the latter is outside the proportional band and the power is saturated (at 0 or 100% for a heat only controller). The proportional band moves downscale to the lower cutback point and waits for the measured value to enter it. It then escorts the measured value with full PID control to the setpoint. In some cases it can cause a 'dip' in the measured value as it approaches setpoint as shown in figure B2.3.2 but generally decreases the time to needed to bring the process into operation.

The action described above is reversed for falling temperature.

If cutback is set to Auto the cutback values are automatically configured to 3 × PB.

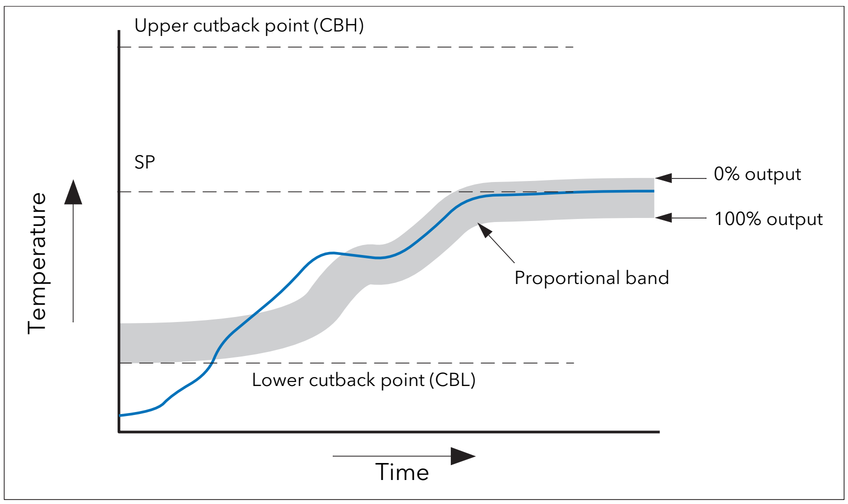



Figure B2.3.2 Cutback operation

Briefly, if PV < CBL then the output is set to its maximum.

If PV > CBH, then the output is set to its minimum

If PV lies within the range CBH-CBL, then PID calculations take control.

### B2.3.3 Manual Reset

With PID control, the integral term automatically removes the steady state error from the setpoint. With PD control, the integral term is set to 'OFF', and the measured value will not settle precisely at the setpoint. The Manual Reset parameter (MR in the PID menu) represents the value of the power output that will be delivered when the error is zero. This value must be set manually in order to remove the steady state error.

**B2.3 LOOP PARAMETERS (Cont.)**

### B2.3.4 Integral Hold

If 'Integral Hold' (Main menu) is set to 'Yes', the integral component of the PID calculation is frozen, that is, it holds its current value but does not integrate any disturbances in the plant. This is equivalent to switching into PD control with a manual reset value preconfigured.

Integral Hold may be used, in a situation where the loop is expected to open. For example, it may be necessary to turn heaters off for a short period or to switch into manual at low power. In this case it may be advantageous to wire Integral Hold to a digital input which activates when the heaters are turned off. When the heaters are switched on again, because the integral is at its previous value, overshoot is minimised.

### B2.3.5 Integral De-bump

This feature is not accessible to the user. When changing from Manual to Auto control. the integral component is forced to: (out put value – proportional component – derivative component) (I = OP – P – D).

This ensures that no change occurs in output at the point of switch over, ('Bumpless Transfer**').** The output power then gradually changes in accordance with the demand from the PID algorithm.

If manual mode = 'Track', bumpless transfer also occurs when changing from Auto to Manual control. At the point of changeover the output power remains the same as the demand in the auto state. The value can then be altered by the operator. For other modes, the output steps to the 'Forced output' or 'Last MOP' value as appropriate. See 'Manual Mode in the Output menu for further details

### B2.3.6 Loop Break

Loop Break attempts to detect loss of restoring action in the control loop by checking the control output, the process value and its rate of change. Since response times vary from process to process, the Loop Break Time (LBT) parameter (PID menu) allows a time to be set before a Loop Break Alarm (Loop Break - Diagnostics menu) becomes active. LBT is set automatically in Autotune.

The Loop Break Alarm parameter has no direct effect on control. In order to define behaviour under Loop Break conditions, the parameter must be wired, for example, to a relay, which can then activate an external indicator.

It is assumed that, so long as the requested output power is within the output power limits of a control loop, the loop is operating in linear control and is therefore not in a loop break condition. If, however, the output becomes saturated then the loop is operating outside its linear control region. If the output remains saturated at the same output power for a significant duration, then this might be symptomatic of a fault in the control loop. The source of the loop break is not important, but the loss of control could be catastrophic.

Since the worst case time constant for a given load is usually known, a worst case time can be calculated over which the load should have responded with a minimum movement in temperature. By performing this calculation the corresponding rate of approach towards setpoint can be used to determine if the loop can no longer control at the chosen setpoint. If the PV was drifting away from the setpoint or approaching the setpoint at a rate less than that calculated, the loop break condition would be met.

If an autotune is performed the loop break time is automatically set to Ti × 2 for a PI or PID loop, or to 12 × Td for a PD loop. For an On/Off controller loop break detection is based on loop range settings as 0.1 × Span where Span = Range High – Range Low. Therefore, if the output is at limit and the PV has not moved by 0.1Span in the loop break time a loop break will occur.

If the loop break time is 0 (off) the loop break time can be set manually. Then, if the output is in saturation and the PV has not moved by >0.5 × Pb in the loop break time, a loop break condition is considered to have occurred.

### B2.3.7 Gain Scheduling

In some processes the tuned PID set may be different at low temperatures from that at high temperatures particularly in control systems where the response to the cooling power is significantly different from that of the heating power, or when changes in the process have occurred. Gain scheduling allows a number of PID sets to be stored and provides automatic transfer of control between one set of PID values and another. For this instrument, the maximum number of sets is three which means that two boundaries are provided to select when the next PID set is used. When a boundary is exceeded the next PID set is selected bumplessly. Hysteresis is used to stop scheduling oscillation at the boundaries.

Gain scheduling is basically a look up table which can be selected using different strategies or types. Auto tune tunes to the active scheduled PID set.

The following Gain Scheduled types are offered using the PID menu parameter 'Sched Type':

| | |
|---|---|
| Set | Required set selected by the user. Alternatively soft wiring may be used to control the PID set selection |
| Setpoint | Transfer between sets is dependent on the setpoint value |
| PV | Transfer between sets is dependent on the process value |
| Error | Transfer between sets is dependent on the Error value |
| Output | Transfer between sets is dependent on the output demand value |
| Remote | A remote parameter may be wired into the scheduler. The PID set is then selected according to the value of this input. |

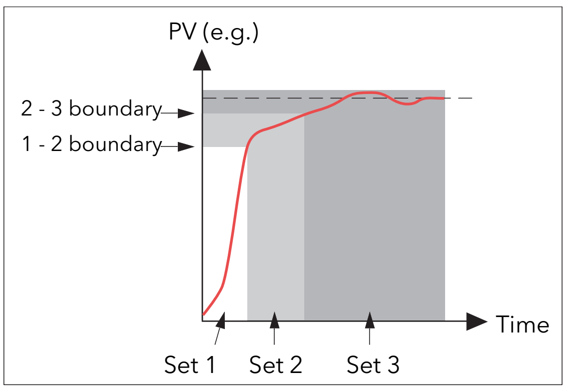



Figure B2.3.7 gain scheduling

## B2.4 TUNING

### B2.4.1 Introduction

The balancing of the P, I and D terms varies from process to process. In a plastics extruder, for example, there are different responses to a die, casting roll, drive loop, thickness control loop or pressure loop. In order to achieve the best performance from an extrusion line all loop tuning parameters must be set to their optimum values.

Tuning involves setting the following PID menu parameters:

Proportional Band (PB), Integral Time (Ti), Derivative Time (Td), Cutback High (CBH), Cutback Low (CBL), and Relative Cool Gain (R2G - applicable to heat/cool systems only).

The recorder/controller is shipped with these parameters set to default values. In many cases the default values give adequate, stable, straight-line control, but the response of the loop may not be ideal. Because process characteristics vary it is often necessary to adjust the control parameters to achieve best control. To determine the optimum values for any particular loop or process it is necessary to carry out a procedure called loop tuning. If significant changes are later made to the process which affect the way in which it responds it may be necessary to retune the loop.

Users have the choice of tuning the loop automatically or manually. Both procedures require the loop to oscillate and both are described in the following sections.

### B2.4.2 Loop Response

Ignoring loop oscillation, there are three categories of loop performance *viz* Under damped, Critically damped and Over damped:

**UNDER DAMPED**

In this situation the parameters are set to prevent oscillation but lead to an overshoot of the Process Value (PV) followed by decaying oscillation until the PV finally settles at the Setpoint. This type of response can give a minimum time to Setpoint but overshoot may cause problems in certain situations and the loop may be sensitive to sudden changes in PV, resulting in further decaying oscillations before settling once again.

**CRITICALLY DAMPED**

This represents an ideal situation where noticeable overshoot to small step changes does not occur and the process responds to changes in a controlled, non oscillatory manner.

**OVER DAMPED**

In this situation the loop responds in a controlled but sluggish manner which results in a non-ideal and unnecessarily slow loop performance.

### B2.4.3 Initial Settings

In addition to the tuning parameters listed above, there are a number of other parameters which can affect loop response. These parameters must be correctly configured before tuning is initiated. Parameters include, but are not limited to:-

**SETPOINT**

Before tuning, the loop conditions should be set as closely as practicable to the actual conditions which will be met in normal operation. For example, in a furnace or oven application a representative load should be included, an extruder should be running, etc.

**OUTPUT HIGH, OUTPUT LOW**

These Output menu heat and cool limits define the overall maximum and minimum power which may be delivered to the process by the control loop. For a heat only controller the default values are 0 and 100%. For a heat/cool controller the defaults are -100 and 100%. Although most processes are designed to work between these limits there may be instances where it is desirable to limit the power delivered to the process.

**REM. OUTPUT LOW, REM. OUTPUT HIGH**

If these Remote Output Limits parameters (Output menu) are used, they are effective only if they lie within the Heat/Cool Limits above.

**CH2 DEADBAND**

Heat/Cool Dead band If a second (cool) channel is configured, a parameter 'Ch2 Deadband' is also available in the Output menu which sets the distance between the heat and cool proportional bands. The default value is 0% which means that heating will cease to be available at the same time as cooling becomes available. The dead band may be set to ensure that there is no possibility of the heat and cool channels operating together, particularly when cycling output stages are installed.

**MINIMUM ON TIME**

If either or both of the output channels is fitted with a relay or logic output, the parameter 'Min On Time' appears in the output menu. This is the cycling time for a time proportioning output and should be set correctly before tuning is started.

**FILTER**

The 'Filter' parameter is found in the Channel 'Main' menu (section 4.4). It is used to remove noise from slowly changing signals so that the underlying trend can be seen more clearly.

### B2.4.3 INITIAL SETTINGS (Cont.)

#### RATE

Sets the maximum PID rate-of-change. The output rate limit is active during tuning and can affect the tuning results. Rate is useful in preventing rapid changes in output from damaging the process or heater elements. The parameter 'Rate' is found in the 'Setpoint' menu.

#### CH1 TRAVEL TIME, CH2 TRAVEL TIME

Valve Travel Time. If the output is a motor valve positioner the 'Ch1 Travel Time' and Ch2 Travel Time' Output menu parameters must be set correctly. The valve travel time is the time taken for the valve to travel from 0% (closed) to 100% (open). This may be different from the motor travel time limits because the mechanical linkage between the motor and the valve, setting of limit switches etc. can modify behaviour. In a valve positioner application, the channel output is wired to the 'PV' input of relay 2A2B or 4AC. Configuring this relay as Type = Valve Raise causes the associated relay (3A3C or 5AC respectively) to be configured automatically as Type = Valve Lower, and the action of the relay pair is controlled by the single wire. In a heat/cool application, channel one is the heat valve and channel two is the cool valve.

### B2.4.4 Other tuning considerations

If a process includes adjacent interactive zones, each zone should be tuned independently with the adjacent zones at operating temperature.

It is recommended that a tuning process be initiated when the PV and setpoint are far apart. This allows start up conditions to be measured and cutback values to be calculated more accurately. Cutback is not set for 'Tune at setpoint'.

In a programmer/controller tuning should only be attempted during dwell periods and not during ramp stages. If a programmer/controller is tuned automatically the controller should be placed in 'Hold' during each dwell period whilst autotune is active.

> Note: Tuning, carried out in dwell periods which are at different extremes of temperature may give different results owing to non linearity of heating (or cooling). This may provide a convenient way to establish values for Gain Scheduling.

If an auto tune is initiated there are two further parameters (High Output' and 'Low Output') which need to be set. These are found in the 'Tune' menu.

| | |
|---|---|
| High Output | Sets a high output limit to be imposed during autotune. Must be ≤ Output High, set in the Output menu. |
| Low Output | Sets a low output limit to be imposed during autotune. Must be ≥ Output Low, set in the Output menu. |

The above values must be set correctly, otherwise sufficient power to achieve SP might not be available during tuning, and the tune will eventually fail.

### B2.4.5 Autotune

Autotune automatically sets the following PID menu parameters:

| | |
|---|---|
| PB | Proportional band. |
| Ti | Integral time. If previously set to 'Off' Ti will remain off after an autotune. |
| Td | Derivative time. If previously set to 'Off' Td will remain off after an autotune. |
| CBH, CBL | Cutback high and low values. If either is set to 'Auto', it will remain so after auto tuning. In order that Autotune set the cutback values for the user, a value other than 'Auto' must be selected before Autotune is initiated. Autotune never returns cutback values less than 1.6 × PB |
| R2G | Calculated only if the unit is configured as Heat/Cool. Following an Autotune, R2G lies between 0.1 and 10. If the calculated value lies outside this range, a 'Tune Fail' alarm is set. |
| LBT | Loop break time. Following an autotune, LBT is set to 2 × Ti (if Ti was not previously set 'Off'), or to 12 × Td (if Ti was previously set to 'Off'). |

### B2.4.5 AUTOTUNE (Cont.)

Autotune can be performed at any time, but normally it is performed only once, during the initial commissioning of the process. However, if the process under control subsequently becomes unsatisfactory (because its characteristics have changed), it may be necessary to tune again for the new conditions.

The auto tune algorithm reacts in different ways depending on the initial conditions of the plant. The explanations given later in this section are for the following example conditions:-

1. Initial PV is below the setpoint and, therefore, approaches the setpoint from below for a heat/cool control loop
2. As above, but for a heat only control loop
3. Initial PV is at the same value as the setpoint (tune at setpoint). That is, within 0.3% of the range of the controller if 'PB Units' (Setup menu) is set to 'Percent', or $\pm$1 engineering unit (1 in 1000) if the 'PB Units' is set to 'Eng'. Range is defined as 'Range High' – 'Range Low' for process inputs or the thermocouple or RTD range defined in section A3 for temperature inputs. If the PV is just outside the range stated above the autotune will attempt a tune from above or below SP.

#### AUTOTUNE AND SENSOR BREAK

When the controller is autotuning and sensor break occurs, the autotune aborts and the controller outputs the sensor break output power 'Sbrk OP' set up in the Output menu. Autotune must be re-started when the sensor break condition is no longer present.

#### AUTOTUNE AND INHIBIT OR MANUAL

If the Loop Inhibit is asserted or the controller is put into Manual Mode, any tune in progress will be aborted and will need to be restart once the condition has been removed. Note that it is not possible to start an autotune sequence if the loop is inhibited or in Manual control.

#### AUTOTUNE AND GAIN SCHEDULING

When gain scheduling is enabled and an autotune is performed, the calculated PID values are written into the PID set that is active, on completion of the tune. Therefore, the user may tune within the boundaries of a set and the values will be written into the appropriate PID set. However, if the boundaries are close (because the range of the loop is not large), then, at the completion of the tune, it cannot be guaranteed that the PID values will be written to the correct set particularly if the schedule type is PV or OP. In this situation the scheduler ('Sched Type') should be switched to 'Set' and the 'active set' chosen manually.

#### INITIAL CONDITIONS

Configure the parameters described in sections B2.4.3 and B2.4.4, above.

Notes:

1. The 'tighter' power limit applies. For example, if 'High Output' is set to 80% and 'Output High' is set to 70% then the output power will be limited to 70%
2. The PV must oscillate to some degree to allow the tuner to calculate the relevant values. The limits must be set so as to allow oscillation about the setpoint.

#### INITIATING THE AUTOTUNE

In the Loop Tune menu for the relevant loop, set 'TuneEn' to 'On'.

## B2.4.5 AUTOTUNE (Cont.)

### EXAMPLE 1: AUTOTUNE FROM BELOW SP (HEAT/COOL)

The point at which Automatic tuning is performed (Tune Control Point) lies just below the setpoint at which the process is normally expected to operate (Target Setpoint). This ensures that the process is not significantly overheated or overcooled. The Tune Control Point is calculated as follows:-

Tune Control Point = Initial PV + 0.75(Target Setpoint – Initial PV).

The Initial PV is the PV measured after a 1 minute settling period (point 'B' in the figure below).

Examples:

If Target Setpoint = 500°C and Initial PV = 20°C, then the Tune Control Point is 380°C.

If Target Setpoint = 500°C and Initial PV = 400°C, then the Tune Control Point is 475°C.

This is because the overshoot is likely to be less as the process temperature approaches the target setpoint.

Figure B2.4.5a shows the auto tune sequence.

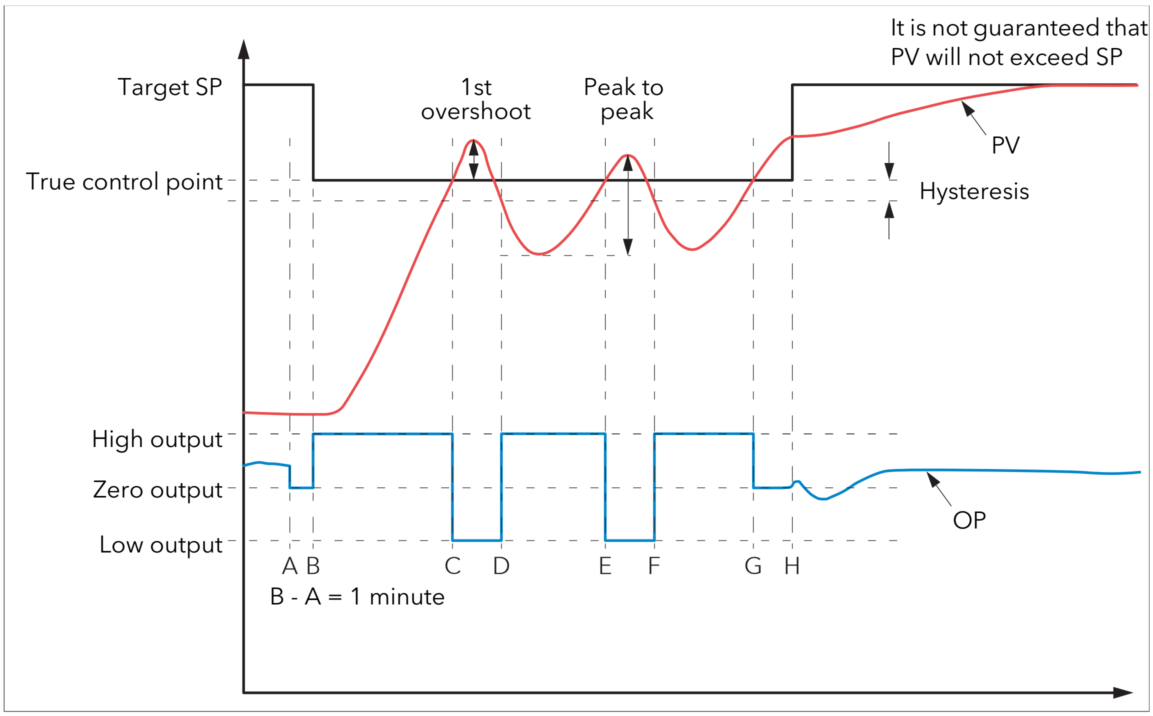



Figure B2.4.5a Autotune heat/cool process

KEY

| | |
|---|---|
| A | Start of Autotune |
| A to B | Heating and Cooling off for one minute allows steady state conditions to be established. |
| B to D | First heat/cool cycle to establish first overshoot. Cutback low (CBL) value calculated from the overshoot magnitude (unless CBL set to 'Auto'). |
| B to F | Two cycles of oscillation allow peak-to-peak value and oscillation period to be determined. PID terms are calculated. |
| F | Heating is switched on. |
| G | Heating (and cooling) are switched off allowing the plant to respond naturally. Measurements over the period F to G are used to calculate the Relative Cool Gain (R2G). Cutback High is calculated from the equation (CBH = CBL × R2G). |
| H | Autotune is turned off and the process is allowed to control at the target setpoint using the new control terms. |

Note: Controlling from above SP is identical except that heating and cooling are reversed.

### B2.4.5 AUTOTUNE (Cont.)

#### EXAMPLE 2: AUTOTUNE FROM BELOW SP (HEAT ONLY)

The sequence of operation for a heat only loop is the same as that described above for a heat/cool loop, except that the sequence ends at ‘F’ since there is no need to calculate ‘R2G’ (R2G is set to 1.0 for heat only processes). At ‘F’ autotune is turned off and the process is allowed to control using the new control terms.

For a tune from below setpoint ‘CBL’ is calculated on the basis of the size of the overshoot (assuming it was not set to Auto in the initial conditions). CBH is then set to the same value as CBL.

> Note: Autotune can also occur when the initial PV is above SP. The sequence is the same as tuning from below setpoint except that the sequence starts with natural cooling applied at ‘B’ after the first one minute settling time. In this case CBH is calculated and CBL is then set to the same value as CBH.

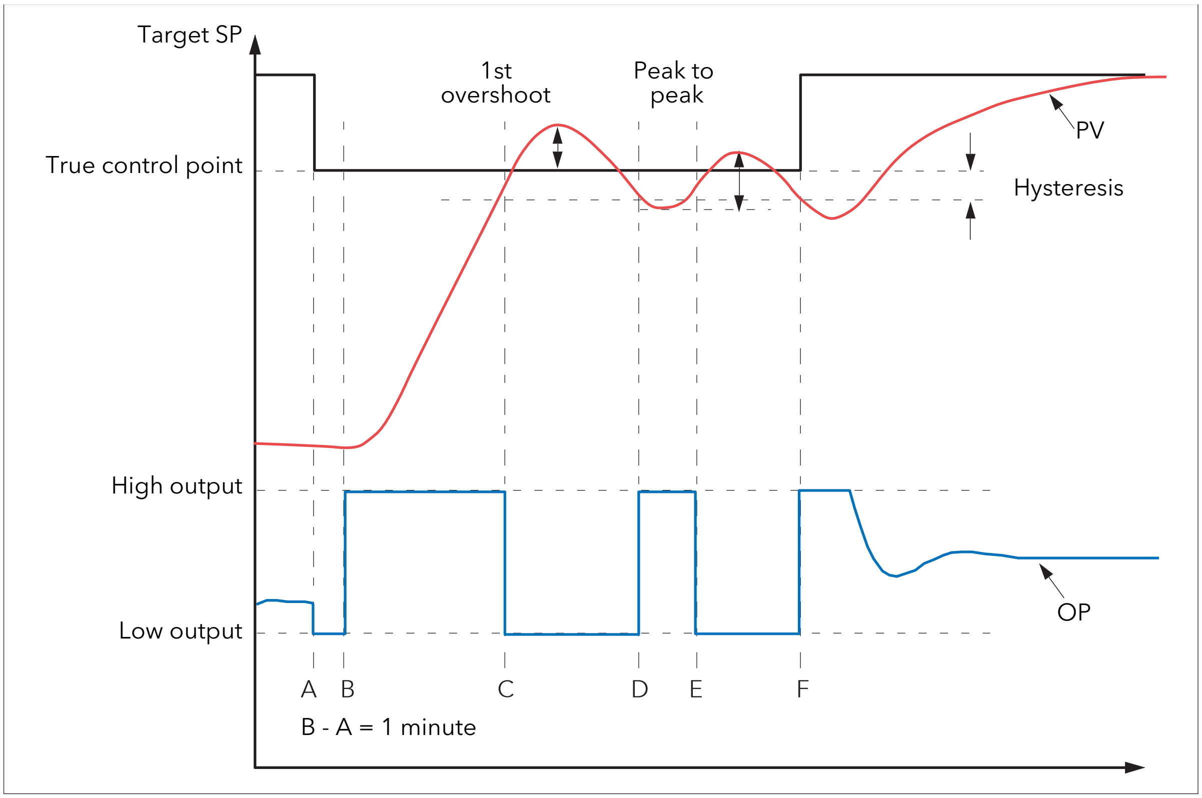



Figure B2.4.5b Autotune heat only process (from below SP)

| | |
|---|---|
| A | Start of Autotune |
| A to B | Heating off for one minute to allow steady state conditions to be established. |
| B to D | First heat cycle to establish first overshoot. Cutback low (CBL) value calculated from the overshoot magnitude (unless CBL set to ‘Auto’). |
| D to F | Calculate PID terms. |
| F | Autotune is turned off and the process is allowed to control at the target setpoint using the new control terms. |

### B2.4.5 AUTOTUNE (Cont.)

### EXAMPLE 3: AUTOTUNE AT SP (HEAT /COOL)

It is sometimes necessary to tune at the actual setpoint being used as shown below.

For a tune at setpoint, autotune does not calculate cutback since there was no initial start up response to the application of heating or cooling. Cutback values of less than 1.6 × PB will not be returned.

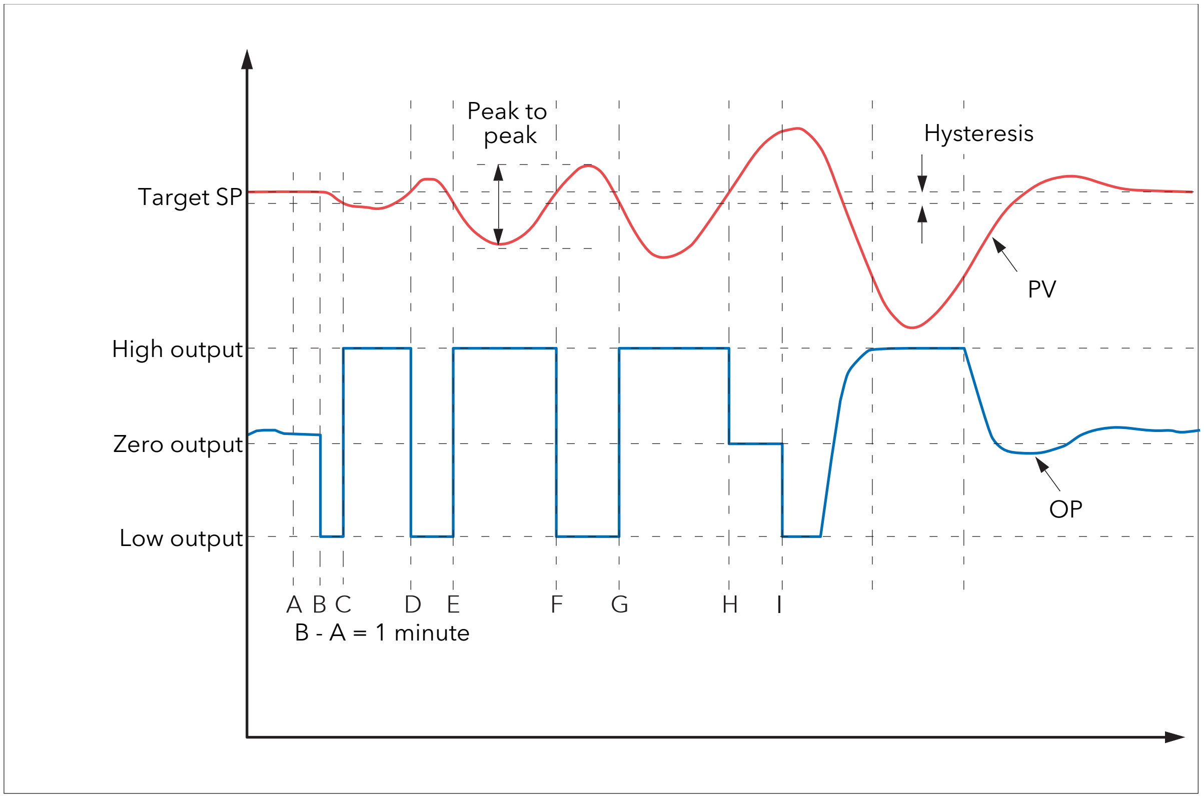



Figure B2.4.5c Autotune at setpoint

| | |
|---|---|
| A | Start of Autotune. A test is done at the start of autotune to establish conditions for a tune at setpoint. Conditions are that SP must remain within 0.3% of the range of the controller if 'PB Units' (Setup menu) is set to 'Percent', or ±1 engineering unit (1 in 1000) if the 'PB Units' is set to 'Eng'. Range is defined as 'Range High' – 'Range Low' for process inputs or the thermocouple or RTD range defined in section A3 for temperature inputs. |
| A to B | The output is frozen at he current value for one minute, and the conditions are continuously monitored during this period. If the conditions specified above are met, then an autotune at setpoint is initiated at 'B'. If PV drifts outside the condition limits at any time during this period, tuning at SP is abandoned, and tuning resumes as a 'tune from above' or 'tune from below', depending on the direction of drift. Since the loop is already at setpoint, a Tune Control setpoint is not calculated; the loop is forced to oscillate about the Target SP. |
| C to G | The process is forced to oscillate by switching the output between the output limits. The oscillation period and the peak-to-peak response are determined, and the PID terms calculated. |
| G to H | An extra heating stage is initiated, then all heating and cooling are switched off at H, allowing the plant to respond naturally. The relative cool gain (R2G) is calculated. |
| I | Autotune is switched off and the process is allowed to control at the target setpoint using the newly calculated terms. |

### B2.4.5 AUTOTUNE (Cont.)

**AT.R2G**

Some load types and process conditions can cause autotune to set an incorrect value for R2G resulting in an instability in the system after an autotune has completed, In such circumstances, the value of R2G should be checked, and if it is low (approaching 0.1) a manual entry should be made as follows:

1. In the Tune menu, set the AT.R2G parameter to 'No'.
2. In the PID menu, enter the new R2G value (calculated as described below)
3. In the Tune menu, enter a value for Low Output, calculated from: Low Output = -High Output x R2G
4. In the Tune menu, set 'TuneEn' On.

R2G CALCULATION

1. In the Main menu, set the controller to Manual mode
2. Turn heating on (limited by the value of 'Output High' in the Output menu) and measure the heating rate ('H' °C/minute).
3. Allow the process to heat to, say, 10% above the setpoint value then turn the heating off and allow the temperature to settle.
4. Turn cooling power on (limited by the value of 'Output Low' in the Output menu) and measure the cooling rate ('C' °C/minute) whilst allowing the temperature to fall below the setpoint value.
5. Calculate the value of R2G from the equation R2G = (H/C) x (Output Low/output High)

Example:

For a measured heating rate (H) of 10°C per min and a measured cooling rate (C) of 25° per minute and with, Output High = 80% and Output Low = 40%, then R2G = (10/25) x (40/80) = 0.4 x 0.5 = 0.2.

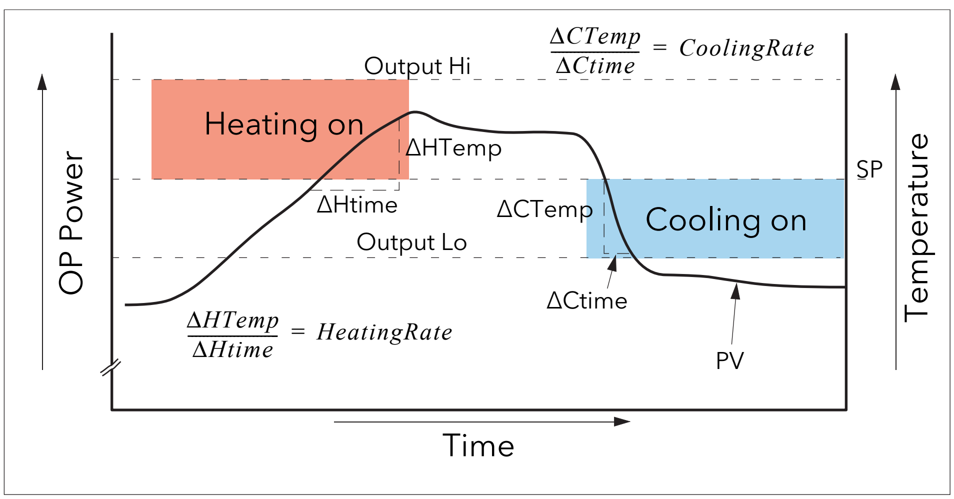



Figure 2.4.5d R2G calculation

Note: This is not a very accurate method as it does not take natural cooling into account. Its main advantage is that it is simple to achieve.

#### B2.4.5 AUTOTUNE (Cont.)

#### FAILURE MODES

The conditions for performing an autotune are monitored by the Tune menu parameter 'State'. If autotune is not successful error conditions are read by this parameter as follows:

| | |
|---|---|
| Timeout | Set if any one stage is not completed within an hour. Possible causes are the loop being open circuit, or not responding to the controller demands. Some heavily lagged systems may produce a timeout if the cooling rate is very slow. |
| TI Limit | This is set if Autotune calculates a value for the integral term which is greater than the maximum allowable (99999 seconds). This indicates that the loop is not responding or that the tune is taking too long. |
| R2G Limit | Error occurs if the calculate value of R2G is outside the range 0.1 to 10.0. R2G limit can occur if the gain difference between heating and cooling is too large, or if the controller is configured for heat/cool, but the heating and/or cooling device is turned off or not working correctly. |

### B2.4.6 Relative Cool Gain in Well Lagged Processes

In the majority of processes Relative Cool Gain R2G is calculated by the autotune algorithm as described in the previous sections.

There are occasions, however, where an alternative algorithm may be preferred. These are processes which are heavily lagged, where the heat loss to ambient is very small so that natural cooling is extremely slow, and certain high order plants, those that need derivative, Td. This algorithm is known as R2GPD and has been added to controllers from firmware version V4.10.

The type of algorithm is selected using the parameter 'Tune R2G' found in the Auto-Tune list, sections 4.6.3 and 4.7.3. The choices are:-

| | |
|---|---|
| Standard | This is the default as described in example 2 in section B2.4.5. and is suitable for use on most processes. The benefit of this algorithm is that it is relatively quick. However, in the type of process described in the previous paragraph, it can produce values which are not ideal. These values are generally identified by R2G equal to or very close to 0.1. |
| R2GPD | If the process is known to be heavily lagged or produces values such as those above then R2GPD should be selected. This algorithm extends the autotune period by putting the controller into proportional plus derivative mode (PD) and uses the output power demand value during this period to determine the Relative Cool Gain. |
| Off | The automatic calculation of Relative Cool Gain can be turned off and the value entered manually as described in section B2.4.6. |

**EXAMPLE 4: WHEN TUNE R2G = R2GPD, AUTOTUNE FROM BELOW SETPOINT**

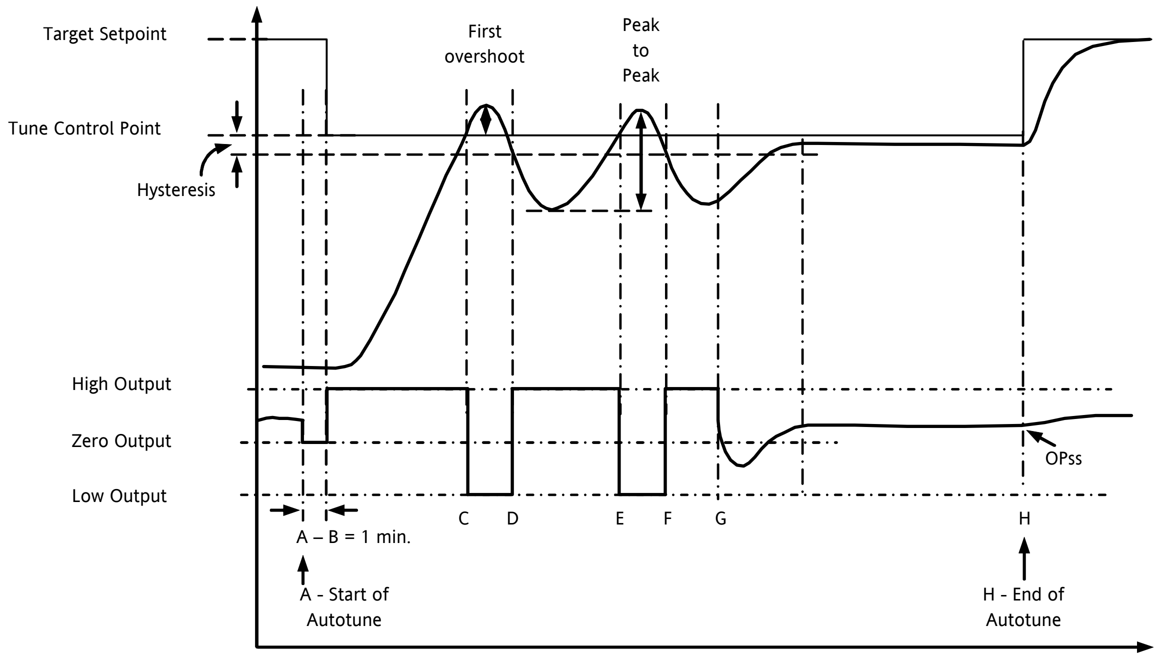



Figure B2.4.6 Autotune from below setpoint

Periods A-F are largely unchanged from the 'Standard' algorithm, example 2 in section B2.4.5 Autotune, with the following exception:

- Changing the Target Setpoint during period A-B will not change the tuning setpoint.

Period F-H is replaced as follows:

| | |
|---|---|
| F to G | Heat is applied for a period (F-G) of half the Ist heat cycle (D-E) to compensate for the last cool cycle. |
| G to H | This is a period in which the controller is put into PD control. The values of proportional term and derivative time for this period of PD control are determined by the algorithm. |
| H | OPss is the output demand value at the end of this period and is used in the determination of R2G. |

## B2.4.7 Manual tuning

If, for any reason, automatic tuning gives unsatisfactory results the controller can be tuned manually. There are a number of standard methods for manual tuning, the Zeigler-Nichols method being described here:

1. Adjust the setpoint to its normal running conditions (assumed to be above the PV so that 'heat only' is applied.
2. Set the integral and derivative times (Ti and Td) to 'Off'
3. Set High and Low cutback (CBH and CBL) to 'Auto'.
4. If the PV is stable (not necessarily at the setpoint), reduce the proportional band (PB) such that the PV just starts to oscillate, leaving time between adjustments to allow the loop to stabilise. Make a note of the PB at this point (PB′), and also note the oscillation period ('T').
   If the PV is already oscillating measure the oscillation period ('T') and then gradually increase PB to the point at which oscillation just ceases. Make a note of the PB (PB′) at this point.
5. If the controller is fitted with a cooling channel, enable this now.
6. Observe the oscillation waveform and adjust 'R2G' until a symmetrical wave form is observed (Figure B2.4.7).

7. Set PB, Ti and Td according to table B2.4.7.

| Control type | PB | Ti | Td |
|---|---|---|---|
| Proportional only | $2 \times PB'$ | Off | Off |
| P + I | $2.2 \times PB'$ | $0.8 \times T$ | Off |
| P + I + D | $1.7 \times PB'$ | $0.5 \times T$ | $0.12 \times T$ |

Table B2.4.7 Calculate parameter values

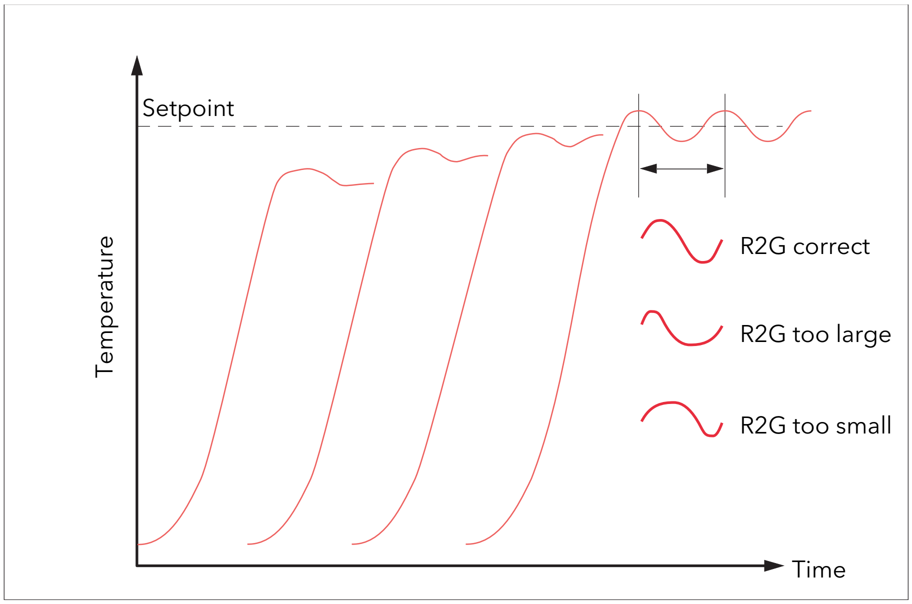



Figure 2.4.6a Relative Cool Gain

**CUTBACK VALUES**

The PID terms calculated from Table 2.4.6, above, should be entered before the cutback values are set.

The above procedure sets up the parameters for optimum steady state control. If unacceptable levels of overshoot or undershoot occur during start-up, or for large step changes in PV, then the cutback parameters should be set manually, as follows:

1. Initially set the cutback values to one proportional bandwidth converted into display units. This can be calculated by taking the value in percent that has been installed into the parameter 'PB' and entering it into the following formula:

   PB/100 × Span of controller = Cutback High and Cutback Low
   For example, if PB = 10% and the span of the controller is 0 to 1200°C, then
   Cutback High = Cutback Low = 10/100 × 1200 = 120

**B2.4.6 MANUAL TUNING (Cont.)**

2. If overshoot is observed following the correct settings of the PID terms increase the value of 'CBL' by the value of the overshoot in display units. If undershoot is observed increase the value of the parameter 'CBH' by the value of the undershoot in display units.

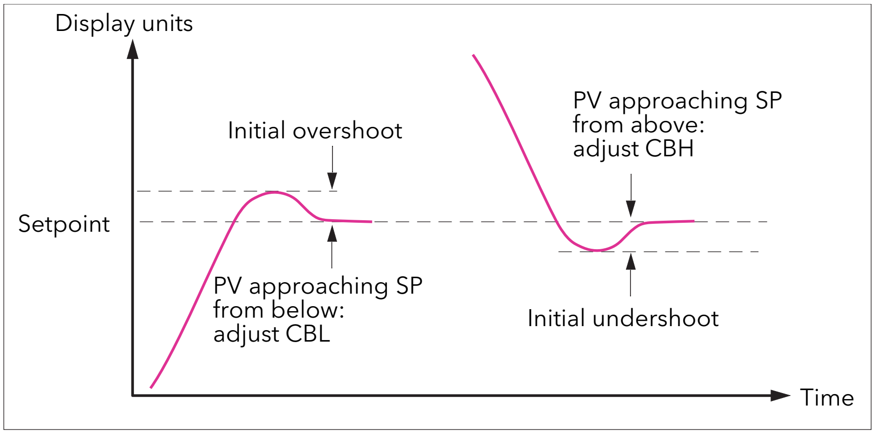



Figure 2.4.6b Manual Cutback setting

## B2.5 SETPOINT

The controller setpoint is the Working Setpoint which can be derived from:-

1. SP1 or SP2, both of which are manually set by the user and can be switched into use by an external signal or via the user interface.
2. From an external (remote) analogue source
3. The output of a programmer function block.

### B2.5.1 Setpoint function block

As well as providing a setpoint, the function block also provides:

1. The ability to limit the rate of change of the setpoint before it is applied to the control algorithm.
2. Upper and lower limits. These are defined as setpoint limits, 'SP High Limit' and 'SP Low Limit', for the local setpoints and instrument range high and low for other setpoint sources.

Note: All setpoints are limited by 'Range High' and 'Range Low' so that if 'SP High Limit', for example, is set higher than 'Range High', then 'SP High Limit' is ignored and the setpoint is limited at the 'Range High' value.

User configurable methods for tracking are available, such that the transfers between setpoints and between operating modes do not cause 'bumps' in the setpoint.

### B2.5.1 Setpoint function block (Cont.)

Figure B2.5.1, below, shows the function block schematic.

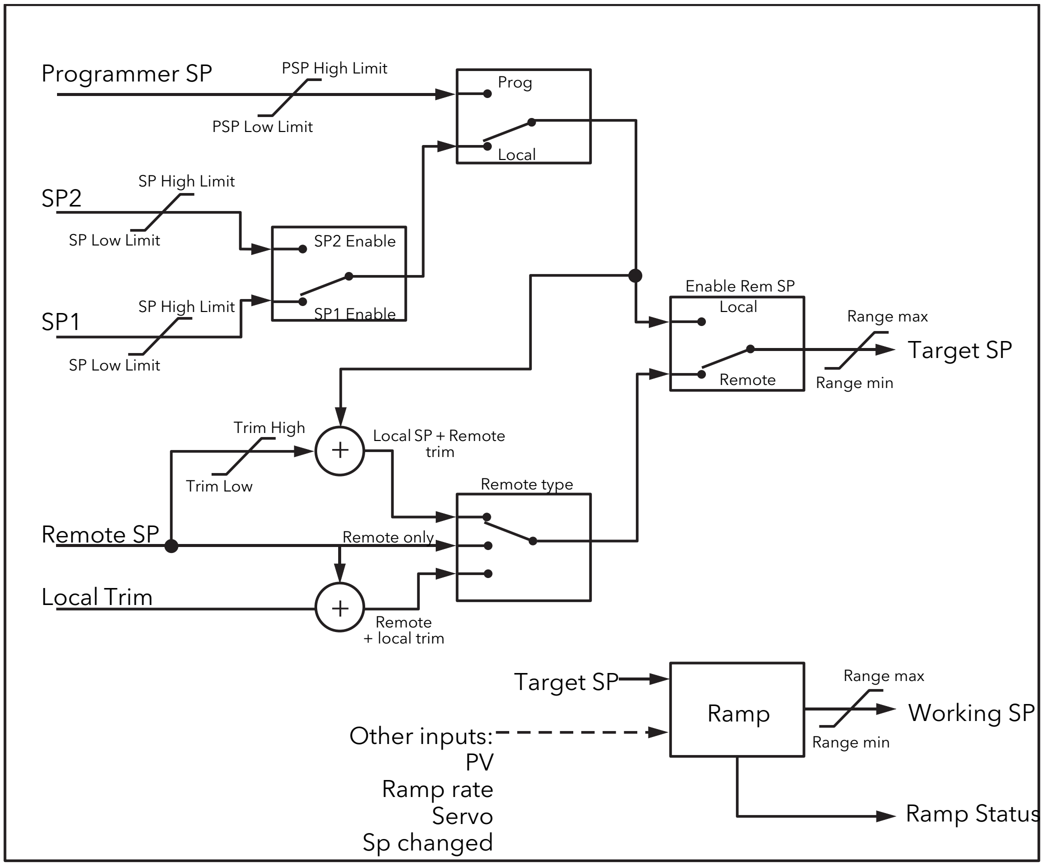



Figure 2.5.1 Setpoint Function block

### B2.5.2 Setpoint Limits

The setpoint generator provides limits for each of the setpoint sources as well as an overall set of limits for the loop. These are summarised in figure 2.5.2, below.

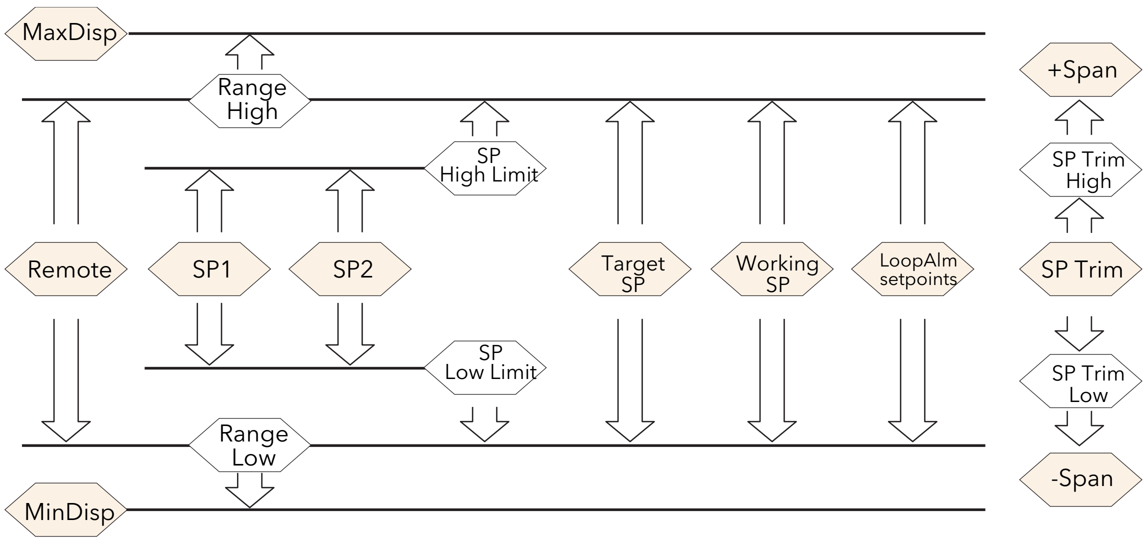



Figure 2.5.2 Setpoint Limits

'Range High' and 'Range Low' provide the range information for the control loop. They are used in control calculations to generate proportional bands. Span = Range High – Range Low.

### B2.5.3 Setpoint Rate Limit

This symmetrical rate limiter allows the rate of change of setpoint to be controlled, preventing step changes in the setpoint. The limit is applied to the working setpoint which includes setpoint trim.

Rate limiting is enabled using the 'Rate' parameter. If this is set to '0' then any change made to the setpoint will be effective immediately. If it is set to any other value, then a change in the setpoint will be have rate limiting applied at the value set, in units per minute. Rate limit applies to SP1, SP2 and Remote SP.

When rate limit is active 'Rate Done' displays 'No'. When the setpoint has been reached the value changes to 'Yes'.

When 'Rate' is set to a value (other than 'Off') an additional parameter 'SP Rate Disable' is displayed which allows the setpoint rate limit to be turned off and on without the need to adjust the 'Rate' parameter between Off and a working value.

If the PV is in sensor break, the rate limit is suspended and the working setpoint takes the value of 0. On sensor break being released the working setpoint goes from 0 to the selected setpoint value at the rate limit.

### B2.5.4 Setpoint Tracking

The setpoint used by the controller may be derived from a number of sources. For example:-

1. Local setpoints SP1 and SP2. These may be selected through the front panel using the parameter 'SP Select', through digital communications or by configuring a digital input which selects either SP1 or SP2. This might be used, for example, to switch between normal running conditions and standby conditions. If Rate Limit is switched off the new setpoint value is adopted immediately when the switch is changed.
2. A programmer generating a setpoint which varies over time. When the programmer is running, the 'Track SP' and 'Track PV' parameters update continuously so that the programmer can perform its own servo. This is sometimes referred to as 'Program Tracking'.
3. From a Remote analogue source. The source could be an external analogue input into an analogue input module wired to the 'Alt SP' parameter or a User Value wired to the 'Alt SP' parameter. The remote setpoint is used when the parameter 'Alt SP Enable' is set to 'Yes'.

Setpoint tracking (sometimes referred to as Remote Tracking) ensures that the Local setpoint adopts the Remote setpoint value when switching from Local to Remote to maintain bumpless transfer from Remote to Local. Bumpless transfer does not take place when changing from Local to Remote.

Note: If Rate Limit is applied, the setpoint will change at the set rate, when changing from Local to Remote.

### B2.5.5 Manual Tracking

When the controller is operating in manual mode the currently selected SP (SP1 or SP2) tracks the PV. When the controller resumes automatic control there will be no step change in the resolved SP. Manual tracking does not apply to the remote setpoint or programmer setpoint.

## B2.6 OUTPUT

### B2.6.1 Introduction

The output function block selects the correct output sources to be used, determines whether to heat or cool and then applies limits. Power feed forward and non-linear cooling are also applied.

It is this block that manages the output in exception conditions such as start up and sensor break.

The outputs, 'Ch1 Output' and 'Ch2 Output', are normally wired to a digital I/O where they are converted into analogue or time proportioned signals for electrical heating, cooling or valve movement.

### B2.6.2 Output Limits

Figure B2.6.2 shows where output limits are applied.

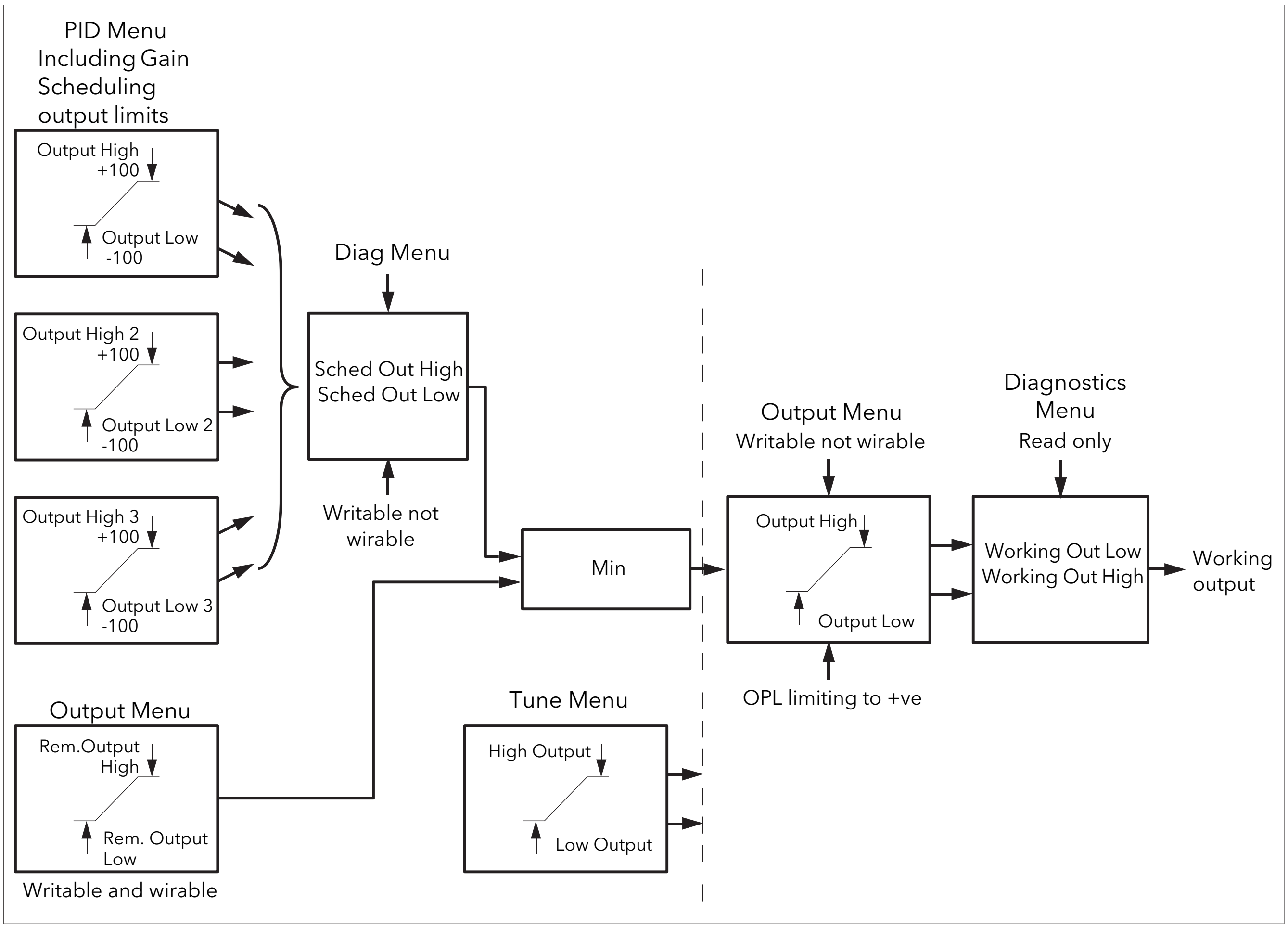



Figure B2.6.2 Output Limits

Notes:
1. Individual output limits may be set in the PID list for each set of PID parameters when gain scheduling is in use.
2. Limits may also be applied from an external source. These are 'Rem.Output High' and 'Rem. Output Low' found in the Output menu. These parameters are wireable; for example they may be wired to an analogue input module so that a limit may applied through some external strategy. If these parameters are not wired ±100% limit is applied every time the instrument is powered up.

(Continued)

### B2.6.2 OUTPUT LIMITS (Cont.)

**Notes (Continued)**

3. The tightest limits (between Remote and PID) are connected to the output where an overall limit is applied using parameters 'Output High' and 'Output Low'.
4. 'Working Out High' and 'Working Out low' found in the Diagnostics list are read only parameters showing the overall working output limits.
5. The tune limits are a separate part of the algorithm and are applied to the output during the tuning process. The overall limits 'Output Hi' and 'Output Lo' always have priority.

### B2.6.3 Output Rate Limit

The output rate limiter is a rate-of-change limiter, set in (%/sec) which prevents step changes in output power being demanded. Rate limiting is performed by determining the direction in which the output is changing, and then incrementing or decrementing the Working Output (Main menu) until it equals the required output (Target OP).

The amount to increment or decrement is calculated using the sampling rate of the algorithm (125ms) and the selected rate limit. If the change in output is less than the rate limit increment the change takes effect immediately.

The rate limit direction and increment is calculated on every execution of the rate limit. Therefore, if the rate limit is changed during execution, the new rate of change takes immediate effect. If the output is changed whilst rate limiting is taking place, the new value takes immediate effect on the direction of the rate limit and in determining whether the rate limit has completed.

The rate limiter is self-correcting such that if the increment is small it is accumulated until it takes effect.

The output rate limit is active when the loop is in both auto and manual modes, and during autotune.

### B2.6.4 Sensor Break Mode

If a Sensor break is detected by the measurement system the loop reacts in one of two ways, according to the configuration of 'Sbrk Mode' ('Safe' or 'Hold'). On exit from sensor break the transfer is bumpless – the power output starts controlling again from the current operating setpoint and moves, under PID closed-loop control, from its pre-set value to the control value.

**SAFE**

If set to 'Safe', the output adopts a pre-set level (Sbrk OP). If rate limit is not configured, the output steps to the Sbrk OP value, otherwise it ramps to this value at the rate limit.

**HOLD**

If set to 'Hold' the output remains at its current value. If Output Rate Limit (Rate) has been configured a small step may be seen as the working output will limit to the value existing two iterations ago.

### B2.6.5 Forced Output

This feature enables the user to specify what the output of the loop should do when moving from automatic control to manual control. The default is that the output power is maintained but it is then adjustable by the user.

If Manual Mode is set to 'Step', the user can set a manual output power value and on transition to manual the output will be forced to that value.

If Manual Mode is set to 'Track' the output steps to the forced manual output and then subsequent edits to the output power are tracked back into the manual output value.

If Manual Mode is set to 'Last Man. Out' then when moving from automatic to manual mode, the output adopts the last manual output value.

### B2.6.6 Power Feed Forward

Power feed forward is used when driving an electrical heating element. It monitors the line voltage and compensates for fluctuations before they affect the process temperature. The use of this will give better steady state performance when the line voltage is not stable.

It is mainly used for digital type outputs which drive contactors or solid state relays. Because it only has value in this type of application it can be switched off using the parameter 'Pff En'. It should also be disabled for any non-electric heating process. It is not necessary when Eurotherm analogue thyristor control is used since compensation for power changes is included in the thyristor driver.

Consider a process running at 25% power, with zero error and then the line voltage falls by 20%. The heater power would drop by 36% because of the square law dependence of power on voltage. A drop in temperature would result. After a time, the thermocouple and controller would sense this fall and increase the ON-TIME of the contactor just enough to bring the temperature back to set point. Meanwhile the process would be running a bit cooler than optimum which may cause some imperfection in the product.

With power feed forward enabled the line voltage is monitored continuously and ON-TIME increased or decreased to compensate immediately. In this way the process need never suffer a temperature disturbance caused by a line voltage change.

'Power Feed forward' should not be confused with 'Feed forward' which is described in section B2.6.8.

### B2.6.7 Cool Type

Cooling methods vary from application to application. For example, an extruder barrel may be cooled by forced air (from a fan), or by circulating water or oil around a jacket. The cooling effect will be different depending on the method. 'Cool Type' (appears only if the 'setup' parameter 'Ch2 Control' is set to 'PID') is used to accommodate different types of cooling methods as follows:

#### LINEAR

The cooling algorithm may be set to linear where the controller output changes linearly with the PID demand signal.

#### OIL COOLING

'Cool Type' = 'Oil'. As oil is, to all intents and purposes, non-evaporative, oil cooling is pulsed in a linear manner.

#### WATER COOLING

If the area being cooled is running well above 100°C, then the first few pulses of water flash into steam giving greatly increased cooling due to the latent heat of evaporation. When the area cools, less (or even no) evaporation takes place and the cooling is less effective.

Setting 'Cool Type' to 'Water' delivers much shortened pulses of water for the first few percent of the cooling range, when the water is likely to be flashing into steam. This compensates for the transition out of the initial strong evaporative cooling.

#### FAN COOLING

'Cool Type = 'Fan'. Fan cooling is much gentler than water cooling and not so immediate or decisive (because of the long heat transfer path through the process mechanics). With fan cooling, a cool gain setting of three upwards is typical. Delivery of pulses to the blower is non linear, this non-linearity being caused by a combination of forced air movement and fan efficiency as a function of air velocity (e. g. the efficiency of a fan when producing a low speed (laminar) air flow is different from its efficiency when producing a high-speed, turbulent flow.

### B2.6.8 Feed forward

Feed forward is a method of adding an extra scalable component to the PID output, before any limiting. It can be used, for example, in the implementation of cascade loops and constant head control or it can be used to pre-load the control signal with a value close to that which is required to achieve the setpoint, thus improving system response. Feed forward (FF) is applied such that the PID output is limited by trim limits and acts as a trim on a FF value. The FF value is derived either from the PV or setpoint by scaling the PV or SP by the 'FF Gain' and 'FF Offset'. Alternatively, a remote value may be used for the FF value, but this is not subject to any scaling. The resultant FF value is added to the limited PID OP and becomes the PID output as far as the output algorithm is concerned. The feedback value then generated must then have the FF contribution removed before being used again by the PID algorithm. The diagram below shows how feed forward is implemented.

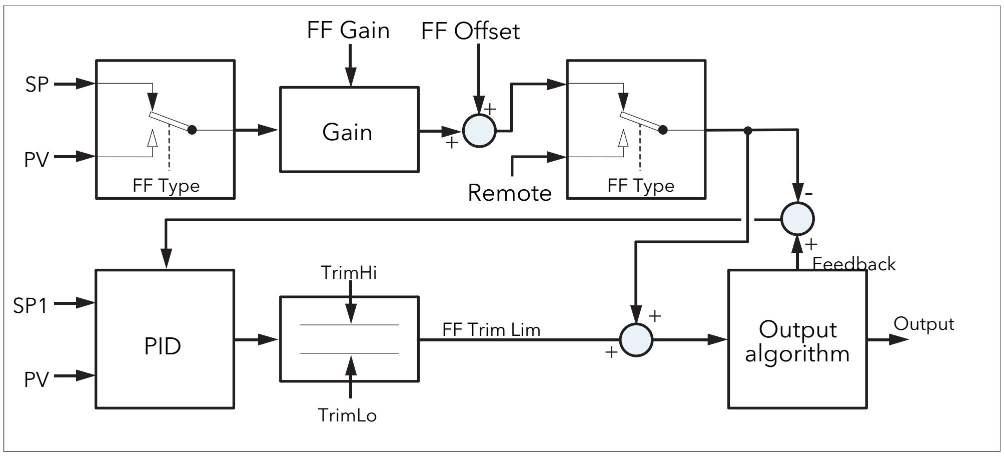



Figure B2.6.8 Implementation of Feed forward

### B2.6.9 Effect of Control Action, Hysteresis and Deadband

#### CONTROL ACTION

For temperature control 'Control Act' should be set to 'Rev'. For a PID controller this means that the heater power decreases as the PV increases. For an on/off controller, output 1 (usually heat) will be on (100%) when PV is below the setpoint and output 2 (usually cool) will be on when PV is above the setpoint.

#### HYSTERESIS

Hysteresis applies to on/off control only and is set in the units of the PV. In heating applications the output will turn off when the PV is at setpoint. It will turn on again when the PV falls below SP by the hysteresis value. This shown in Figures B2.6.9a and B2.6.9b below for a heat and cool controller.

Hysteresis is intended to prevent the output from repeated switching on and off 'chattering' at the control setpoint. If the hysteresis is set to 0 then even the smallest change in the PV when at setpoint will cause the output to switch. Hysteresis should be set to a value which provides an acceptable life for the output contacts, but which does not cause unacceptable oscillations in the PV.

If this performance is unacceptable, it is recommended that PID control be used instead.

#### DEADBAND

Deadband 'Ch2 Deadband' can operate on both on/off control or PID control where it has the effect of extending the period when no heating or cooling is applied. In PID control the effect is modified by both the integral and derivative terms. Deadband might be used in PID control, for example, where actuators take time to complete their cycle thus ensuring that heating and cooling are not being applied at the same time. Deadband is likely to be used, therefore, in on/off control only. Figure B2.6.9b, below, adds a deadband of 20 to the first example in figure B2.6.9a.

## B2.6.9 EFFECT OF CONTROL ACTION, HYSTERESIS AND DEADBAND (Cont.)

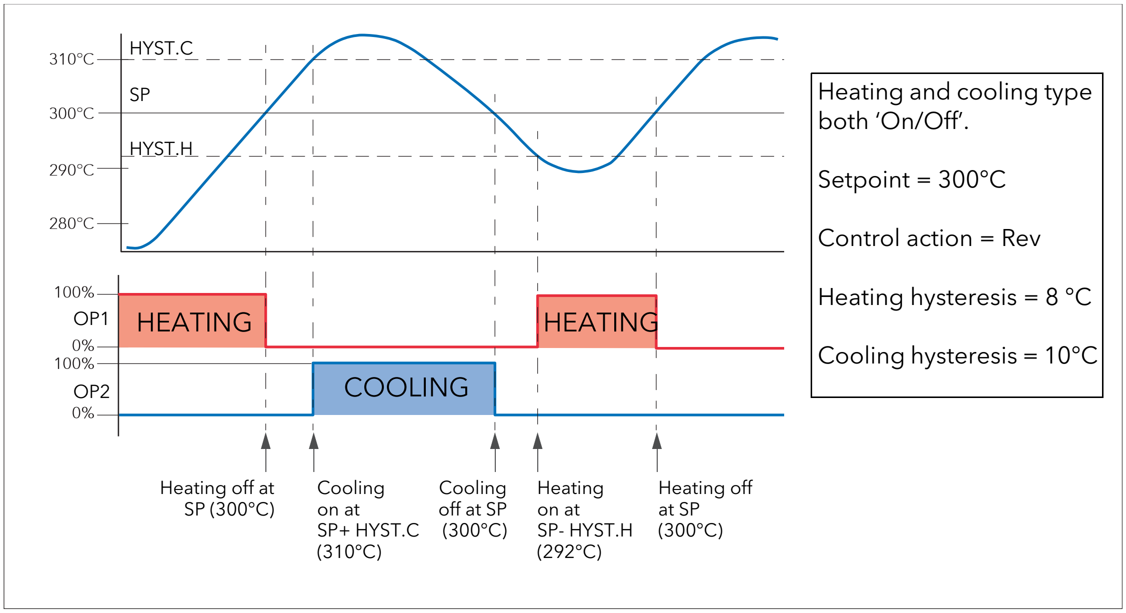



Figure B2.6.9a Deadband OFF

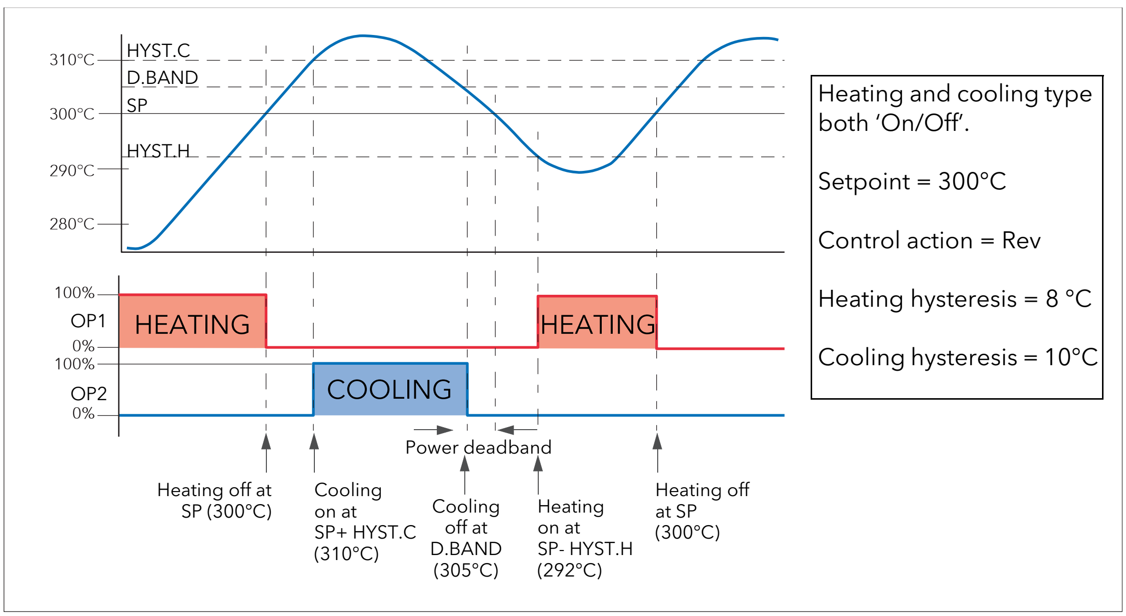



Figure B2.6.9b Deadband ON set at 50% of Cooling.

### B2.6.10 Valve nudge

For systems configured as Unbounded Valve Positioning (VPU) - set up in Loop Setup configuration Ch1(2) control), it is possible to move the valve in small increments towards the open position (Nudge Raise) or towards the closed position (Nudge Lower). The trigger for such nudging can be a digital input (e.g. contact closure) 'wired' to the nudge raise or lower parameter, the up or down arrow keys or a command received over the serial link.

The nudge command causes the valve drive output to drive the valve for either the minimum on time, or for as long as the command is 'true', whichever is the longer (note 2). The default minimum on time is 125ms, but this can be edited in the configuration for the relevant output relay (section 4.11.2).

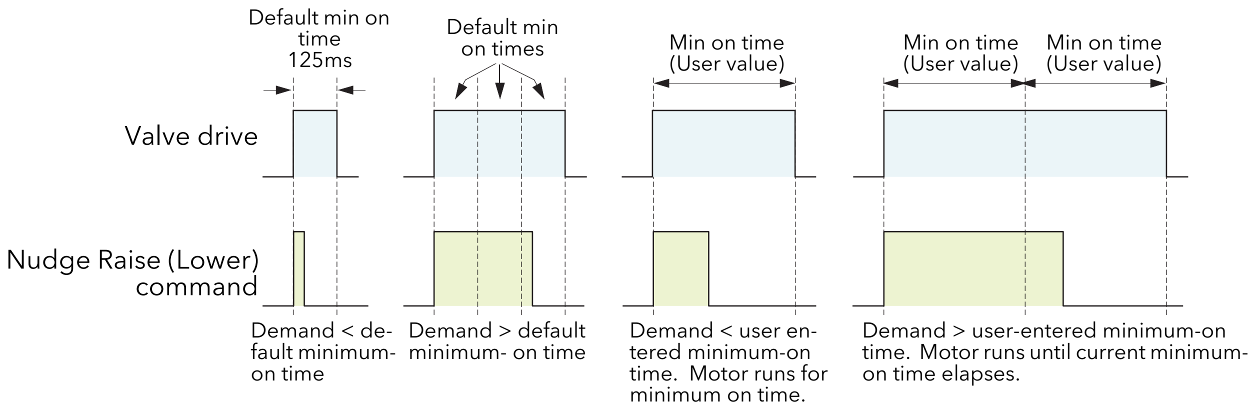



Figure B2.6.10 Valve nudge examples

Notes:

1. If Ch1 is set to VPU, Nudge operates the channel 1 valve, no matter what Ch2 is set to. If Ch1 is not set to VPU, and Ch2 is set to VPU then the nudge operates on channel 2 valve.
2. The minimum on time is continuously retriggered. This means that if a minimum on time of (say) 10 seconds has been configured, then the valve may continue to move for up to 10 seconds after the command has been removed. That is, it continues until the current minimum on time period has expired.

### B2.6.11 Time Proportioning

PID controllers somtimes use Time Proportioning to control the average power to the load. This is done by repeatedly switching the output on for a period ($T_{on}$) and then off for a period ($T_{off}$). The total period ($T_{on}$ + $T_{off}$) is called the ‘cycle time’. During each cycle, the average power delivered to the load is:

$P_{Avg} = P_{Heater}$ x Duty cycle,

where ‘$P_{Heater}$’ is the actual transferred heater (or cooler) power and Duty cycle = $T_{on}/(T_{on} + T_{off})$, normally represented as a percentage value.

The PID controller calculates the Duty Cycle (the PID output control signal from 0 to 100%) and provides a Minimum on time between 100ms to 150 seconds.

Figure B2.6.11 shows how $T_{on}$, $T_{off}$ and cycle time vary with demand %.

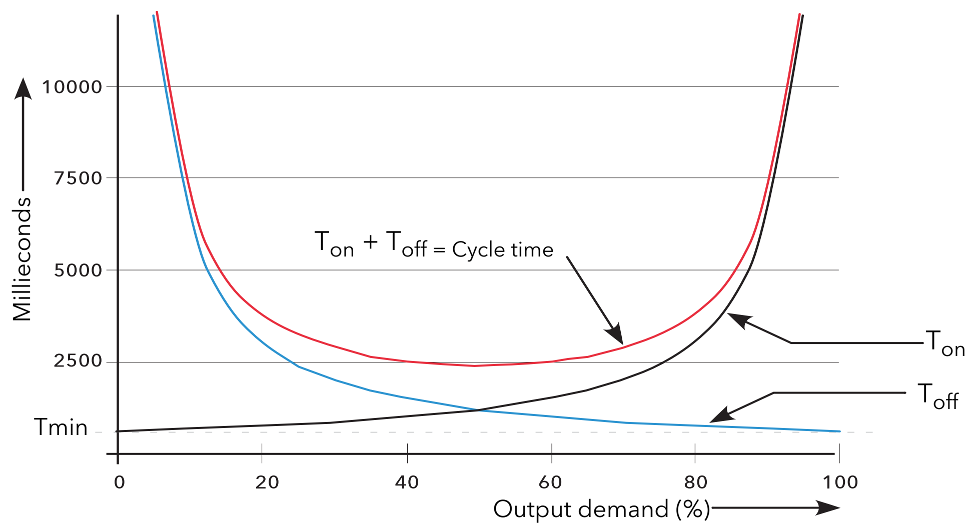



Figure B2.6.11 Time proportioning curves (Minimum on time = 625ms)

Note: For this instrument, only ‘Min on time’ is configurable

## B2.7 DIAGNOSTICS

See section 4.6.7 for definitions of these parameters

# Appendix C: REFERENCE

## C1 BATTERY REPLACEMENT

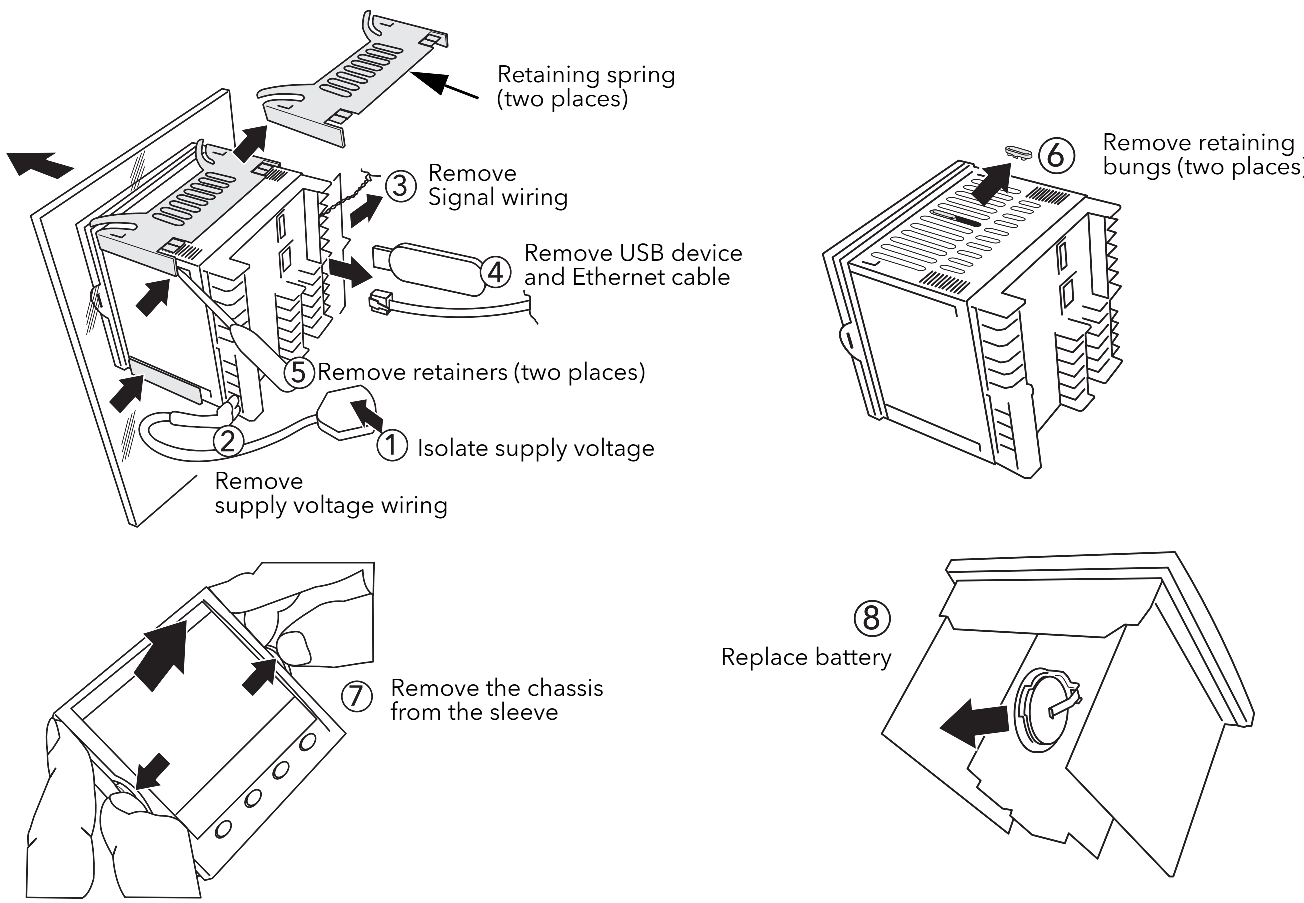
The battery can be replaced only after the unit has been withdrawn from the panel. It is therefore normally necessary to unwire the instrument before changing the battery.

**WARNING**

Before removing the supply voltage wiring, isolate the supply voltage and secure it against unintended operation.

Note: The new battery must be installed within 10 seconds of the exhausted battery's removal, or data will be lost.

1. Isolate the supply voltage and secure it against accidental operation.
2. Remove supply voltage wiring from the rear terminals.
3. Remove all signal wiring
4. Remove the Ethernet cable and USB device if fitted.
5. Remove the two securing springs, using a small screwdriver if necessary.
6. Prise the two chassis-retaining bungs, using a small screwdriver if necessary.
7. Ease the latching ears outwards, whilst pulling forwards on the bezel, until the chassis is free of the sleeve.
8. Replace the battery. Recycle the exhausted battery according to local procedures.
9. Reinsert the chassis into the sleeve, and secure it using the chassis-retaining bungs previously removed.
10. Reinstall the chassis into the panel and secure it using the retaining springs previously removed.
11. Reinstall all wiring, the Ethernet cable and USB device, if any.
12. Reset the date and time as described in section 4.1.1.



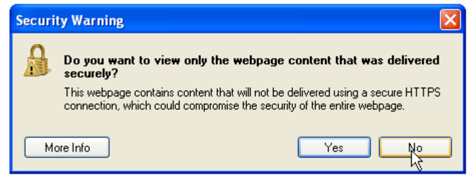
# C2 SETTING UP AN FTP SERVER USING FILEZILLA

## C2.1 DOWNLOADING

'FileZilla' is a free download from the internet (search for 'FileZilla server download').

1. Download the latest version, following the instructions on the screen.

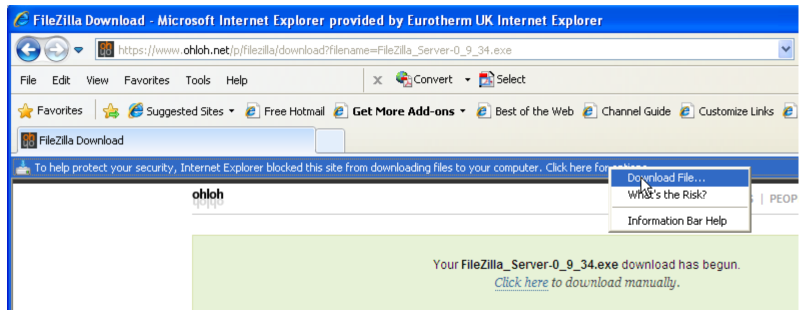
2. Answer 'No' to the question 'Do you want to view only the webpage content that was delivered securely'.

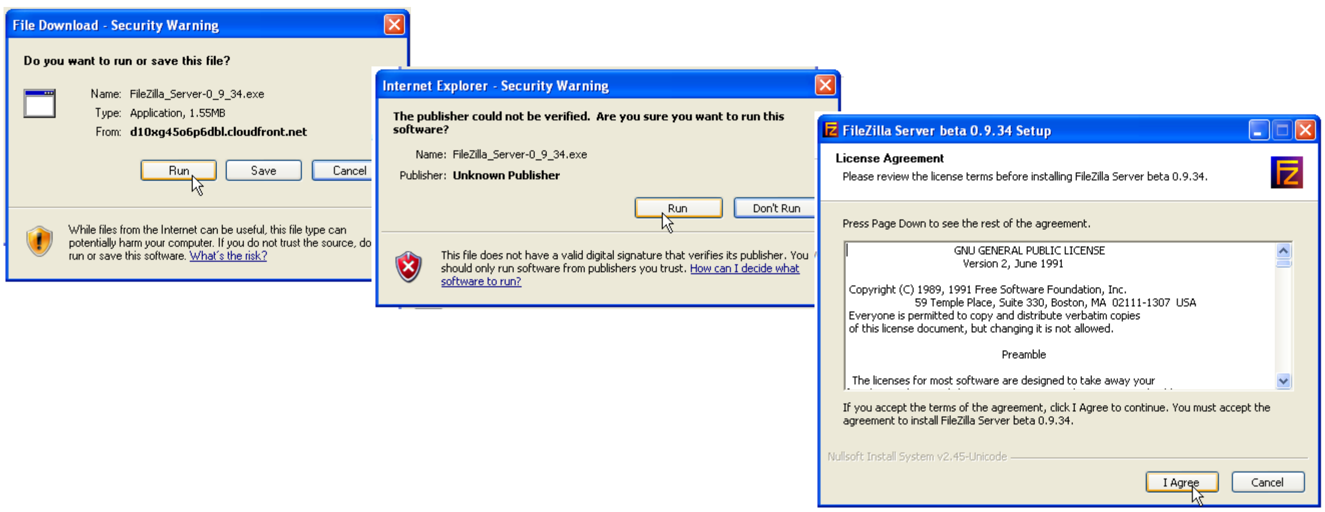


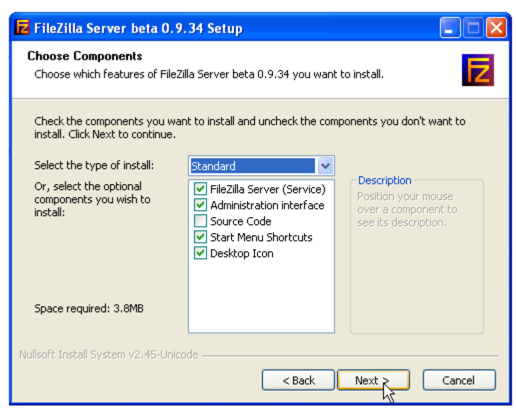

3. If necessary enable file download.

4. In the 'Do you want to run or save this file' Security Warning window- click on 'Run'

5. In the 'The Publisher could not be verified..., Security Warning window, click on 'Run'



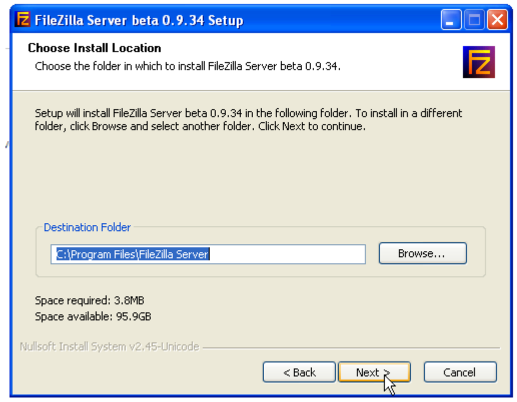



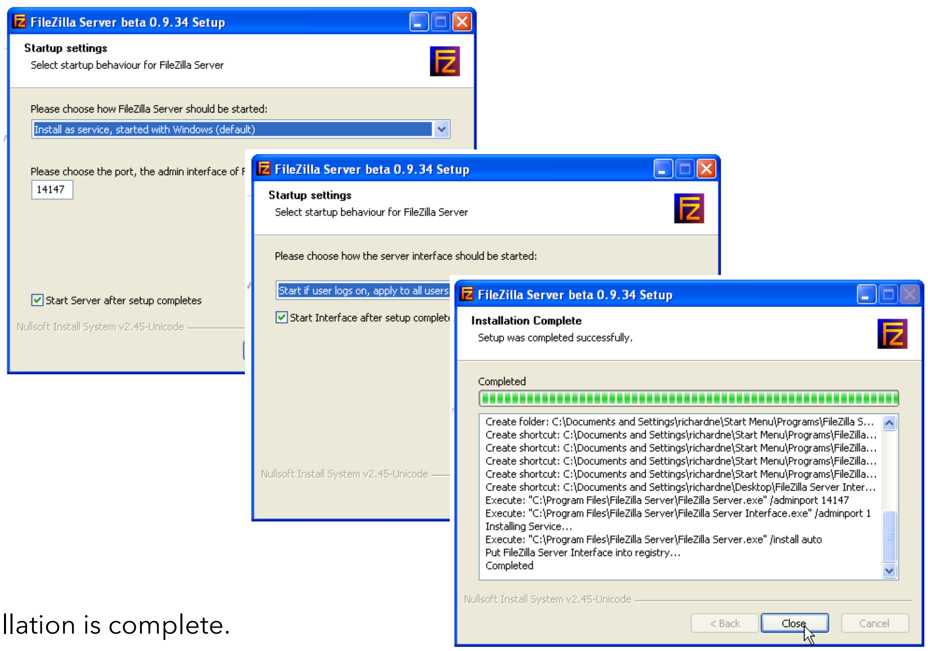

6. Agree or cancel the License agreement. If 'Agree', choose 'Standard' as the type of install.

## C2.1 DOWNLOADING (Cont.)

7. Choose the destination for the file

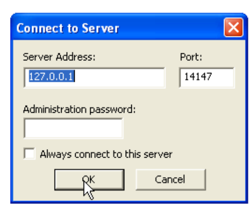



8. Select startup settings



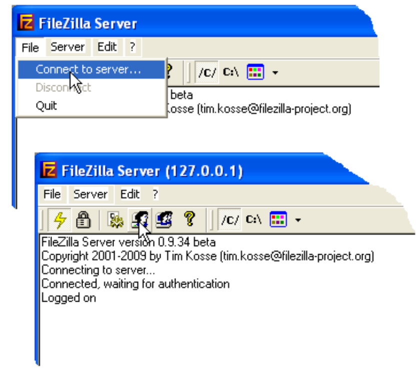

9. Click on Close when Installation is complete.

10. Click ‘OK’ in the ‘Connect to Server’ window.



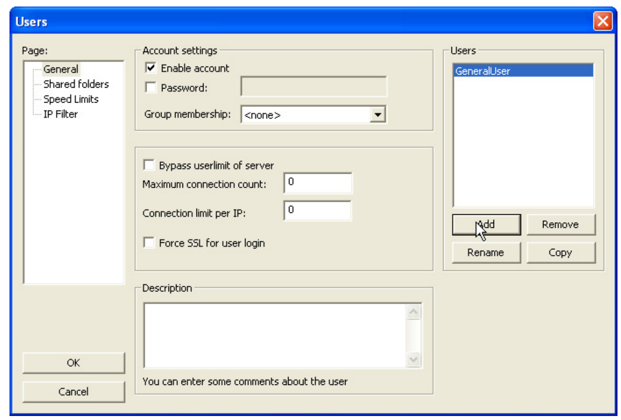
## C2.2 SERVER SETUP

1. Create a new folder (directory) called, for this example, ‘Archive’ in a suitable location such as the C drive, or the desktop.

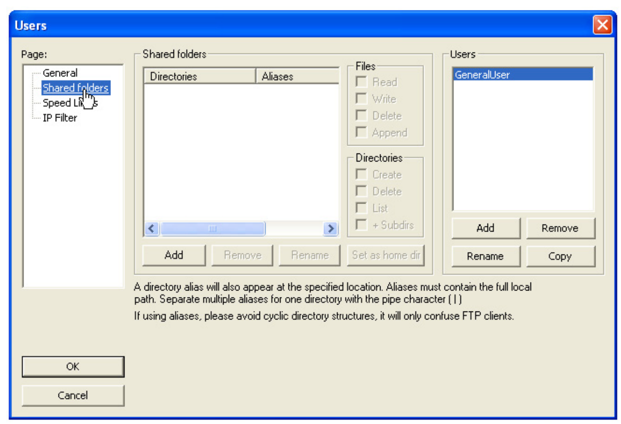



2. In the Filezilla server window, click on ‘File’ and select ‘Connect to Server’.

   The ‘Logged on’ message appears



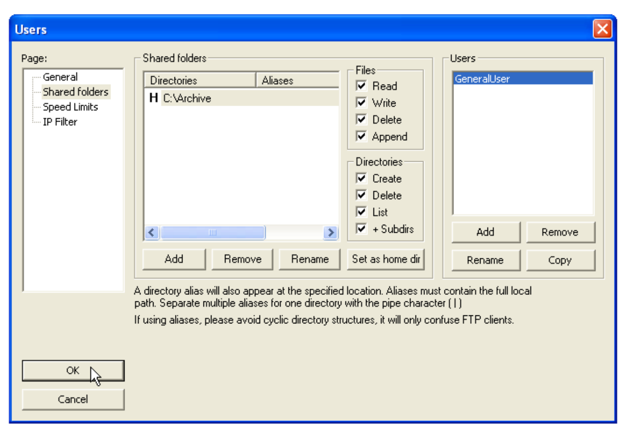

3 In the Edit menu, select ‘Users’ and in the ‘General’ page, click on ‘Add’ and enter a name for the user, then click ‘OK’. For this example, ‘GeneralUser’ has been used, but it may be more advantageous to use ‘Anonymous’ because this is the default name in the recorder/controller. Click on ‘OK’.




4 In the Edit menu, select ‘Users’ and in the ‘Shared Folders’ page, click on ‘Add’

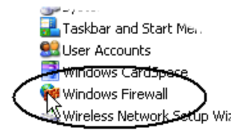
   A browse window opens allowing the user to select the new folder (‘Archive’) created in step 1, above.

   When OK has been clicked to confirm the selection, the new folder appears in the centre window (with an ‘h’ next to it to indicate that this is the home folder for this ftp user setup.



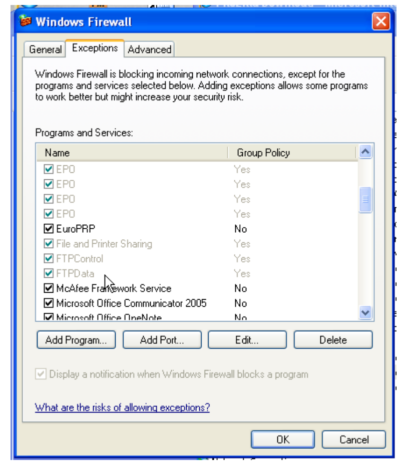

5. Click on the relevant folder to enable the tick boxes. Click on all the ‘File’ and ‘Directory’ enable tick boxes, then click OK

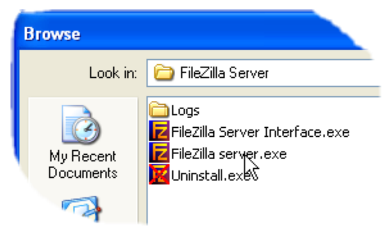
## C2.3 PC SETUP




1. Operate the ‘Start’ button, and select ‘Control Panel’ from the window that appears. Double click on ‘Windows Firewall’

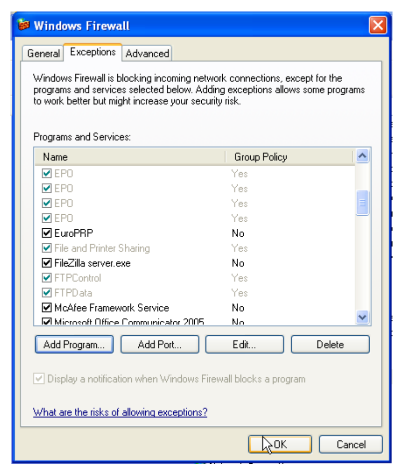
2. Click on the ‘Exceptions’ tab in the window that appears, and check that both ‘FTPControl’ and ‘FTPData’ are enabled (ticked). If not, the user’s IT department should be contacted for advice.




3. Click on ‘Add Program...’ and browse to the Filezilla destination defined in step 7 of the download section (C2.1). Select ‘FileZilla server.exe’ and click on ‘Open’

   ‘FileZilla server.exe’ appears in the Exceptions list.




Click on ‘OK’

## C2.4 RECORDER/CONTROLLER SET UP

In Network Archiving (section 4.2.2):

1. Enter the IP address of the pc in which the FTP server has been enabled in the ‘Primary Server’ field.
2. Enter the Primary User name, as entered in step three of the Server setup procedure (section C2.2) above (GeneralUser in this example).
3. Enter the IP address of another suitable pc which has been configured as an ftp server in the ‘Sec. Server’ field, and enter the relevant ‘Sec. User’ name.
4. Configure the other unattended archive parameters as required (section 4.2.2).

Note: For the example above, ‘Password’ was not enabled in the User Accounts setup page (section C2.2), so for this example any Primary (Sec.) password entry is ignored. If a password had been entered in the User Accounts setup, then the Primary (Sec.) Password field would have to contain this password.

## C2.5 ARCHIVE ACTIVITY

Once a demand or unattended archive is initiated, the FileZilla Server page shows the activity status as the archive progresses. Figure C2.5 shows a typical page. The top of the page shows the transaction details between the server and any clients to which it is connected. The bottom portion shows details of the files currently being transferred. These files are archived to the 'Archive' folder.

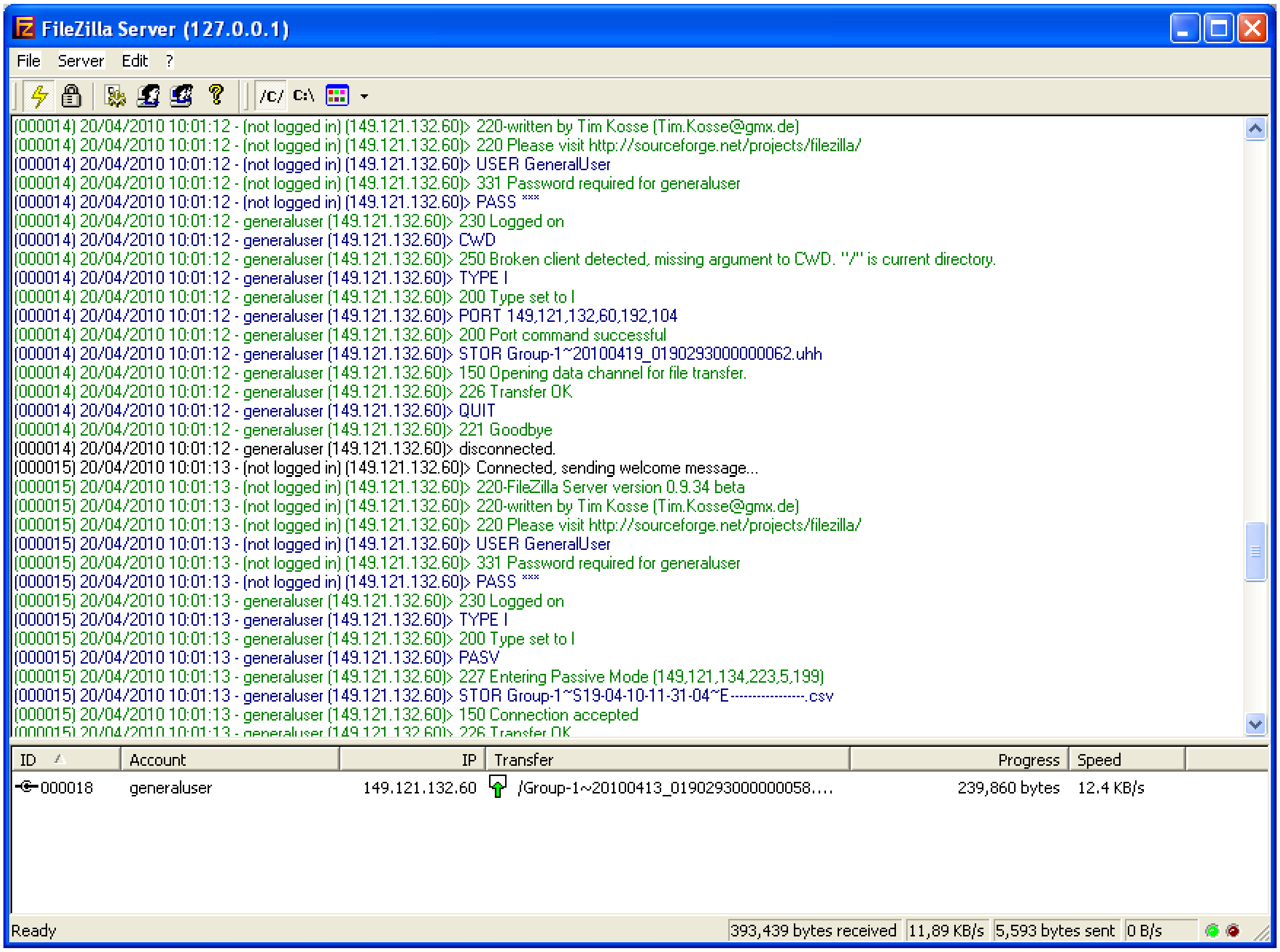


Figure C2.5 FileZilla Server archive activity page

# C3 FUNCTION BLOCK DETAILS

## C3.1 EIGHT INPUT OR BLOCK

An eight input logical OR block whose output is high (1, On) if any one or more inputs is high (1, On). If more than eight inputs are required, a second block is automatically introduced, as shown in figure C3.1a. The blocks in the figure are given the names 'A' and 'B', where 'A' and 'B' can be any of the 12 available instances.

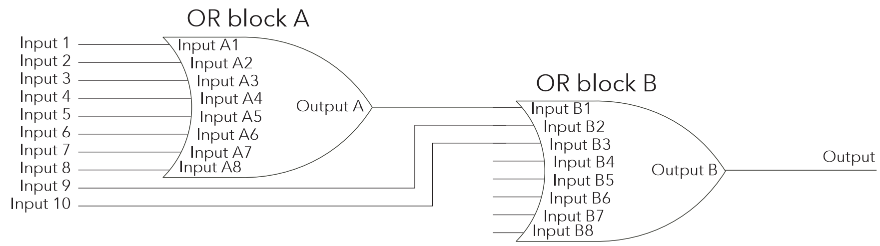



Figure C3.1a Eight input OR block

OR blocks are used automatically by the 'user wiring' when more than one source is wired to the same destination parameter. For example, it may be required that Relay (Digital I/O 2A2B) is to operate if channel 1 alarm 1 and/or channel 2 alarm 1 channels goes active. In such a case, the 'Active' parameter for the two channel alarms would be wired to the same relay's 'PV' parameter.

OR blocks are invisible to the user interface, but the iTools graphical wiring page for this configuration (figure C3.1b), shows that an OR block has been introduced to OR the two alarm outputs together.

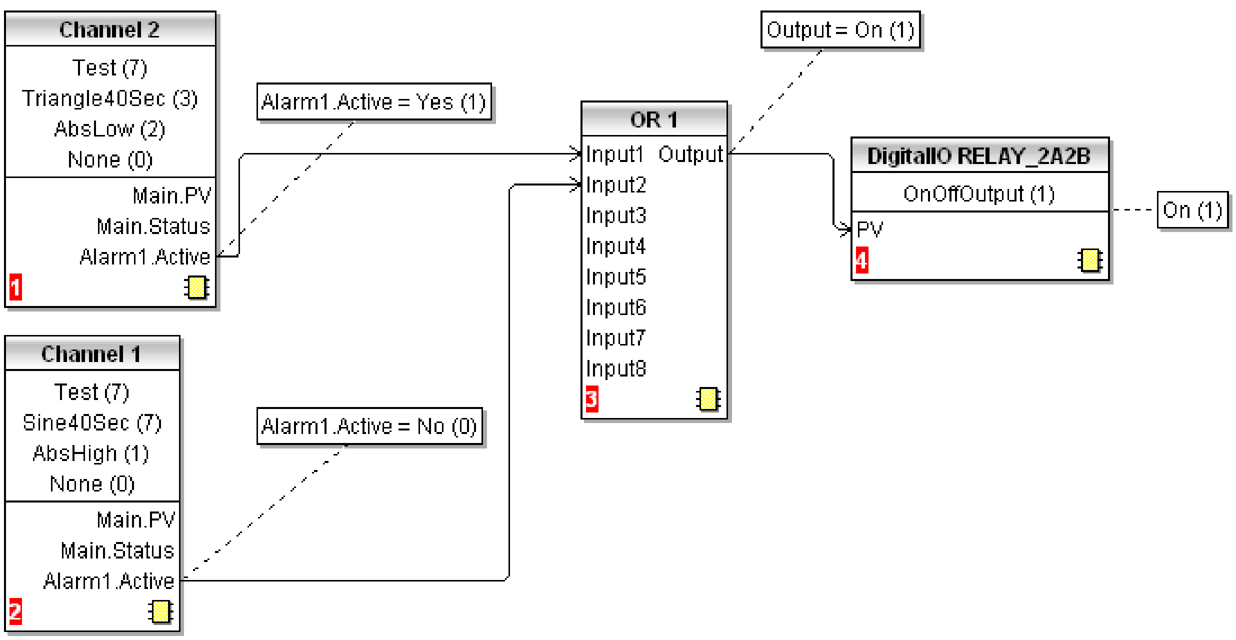



Figure C3.1b iTools representation of OR block usage

## C4 TCP PORT NUMBERS

The following TCP ports are made use of by the instrument.

| Port | Usage |
| --- | --- |
| 20 | File Transfer protocol (FTP) data |
| 21 | FTP control |
| 502 | Modbus TCP communications |

## C5 ISOLATION DIAGRAM

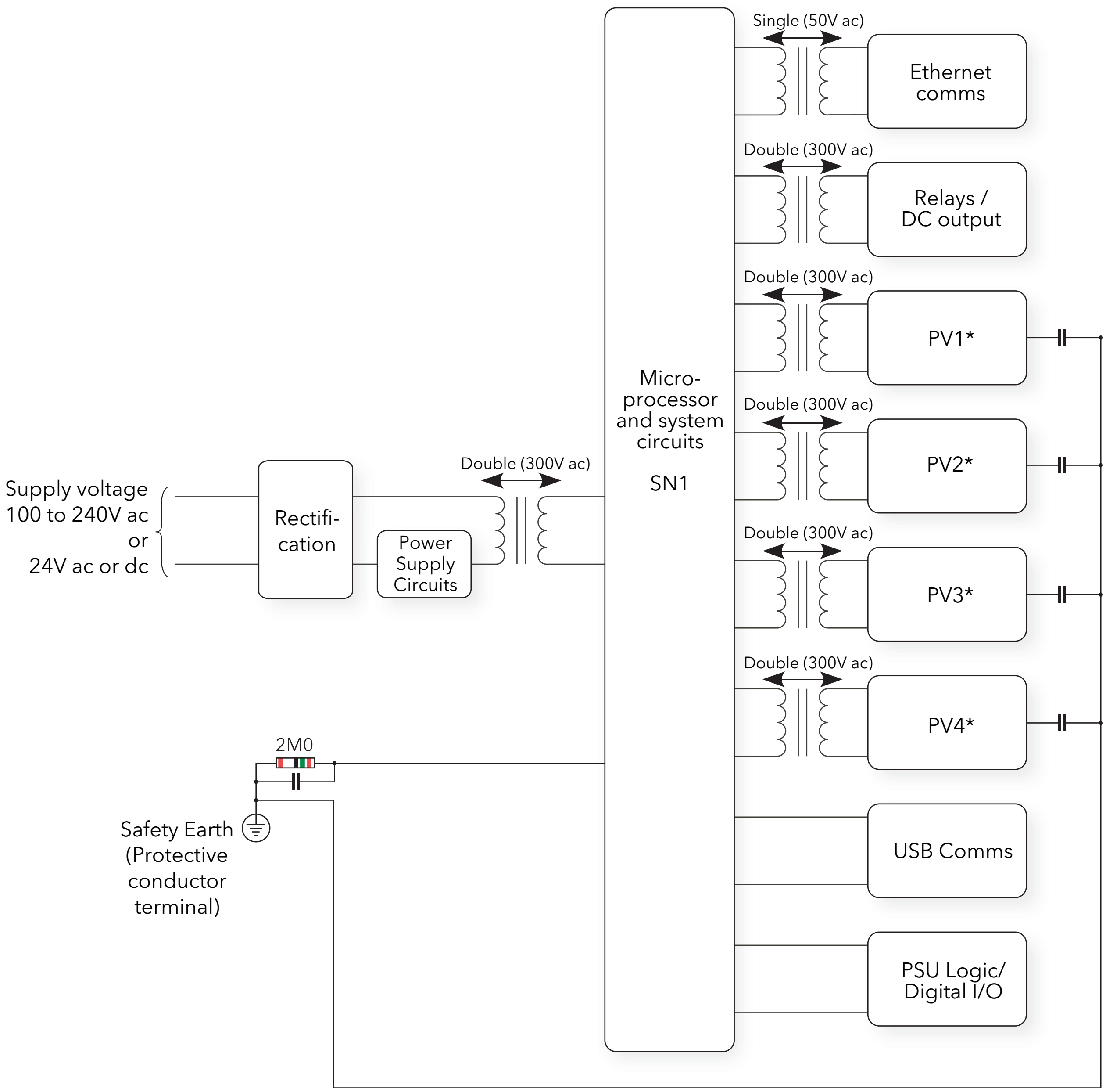



Figure C5 Isolation diagram

* Note: Each 'PV' is double isolated (300VRMS) from all other 'PV's.

## Appendix D:  CONFIGURATION MENU OVERVIEW

This appendix contains an overview of the configuration menus for the instrument, including all options as follows:

## D1 INSTRUMENT CONFIGURATION MENUS

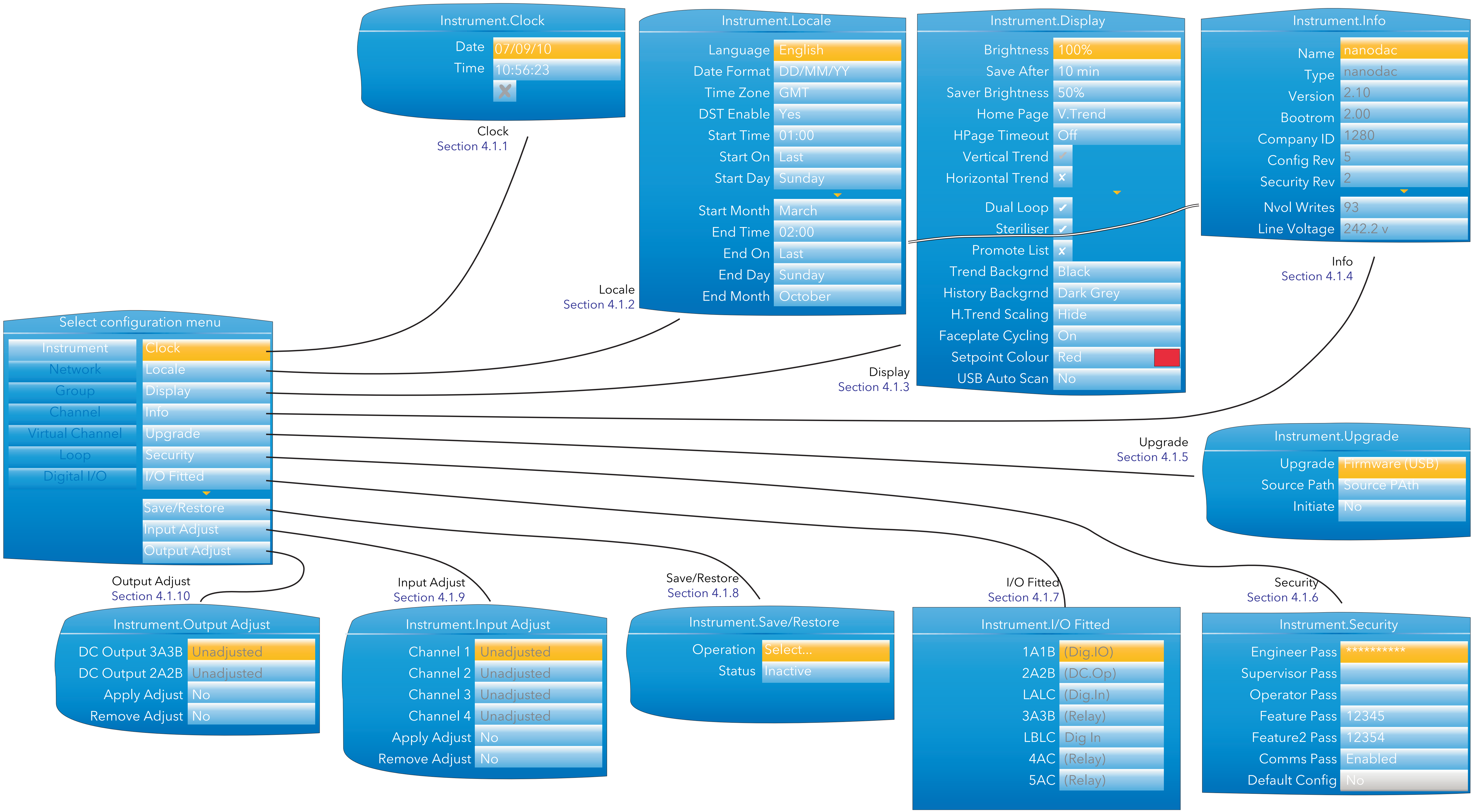



Figure D1 Instrument configuration menus

## D2 NETWORK CONFIGURATION MENUS

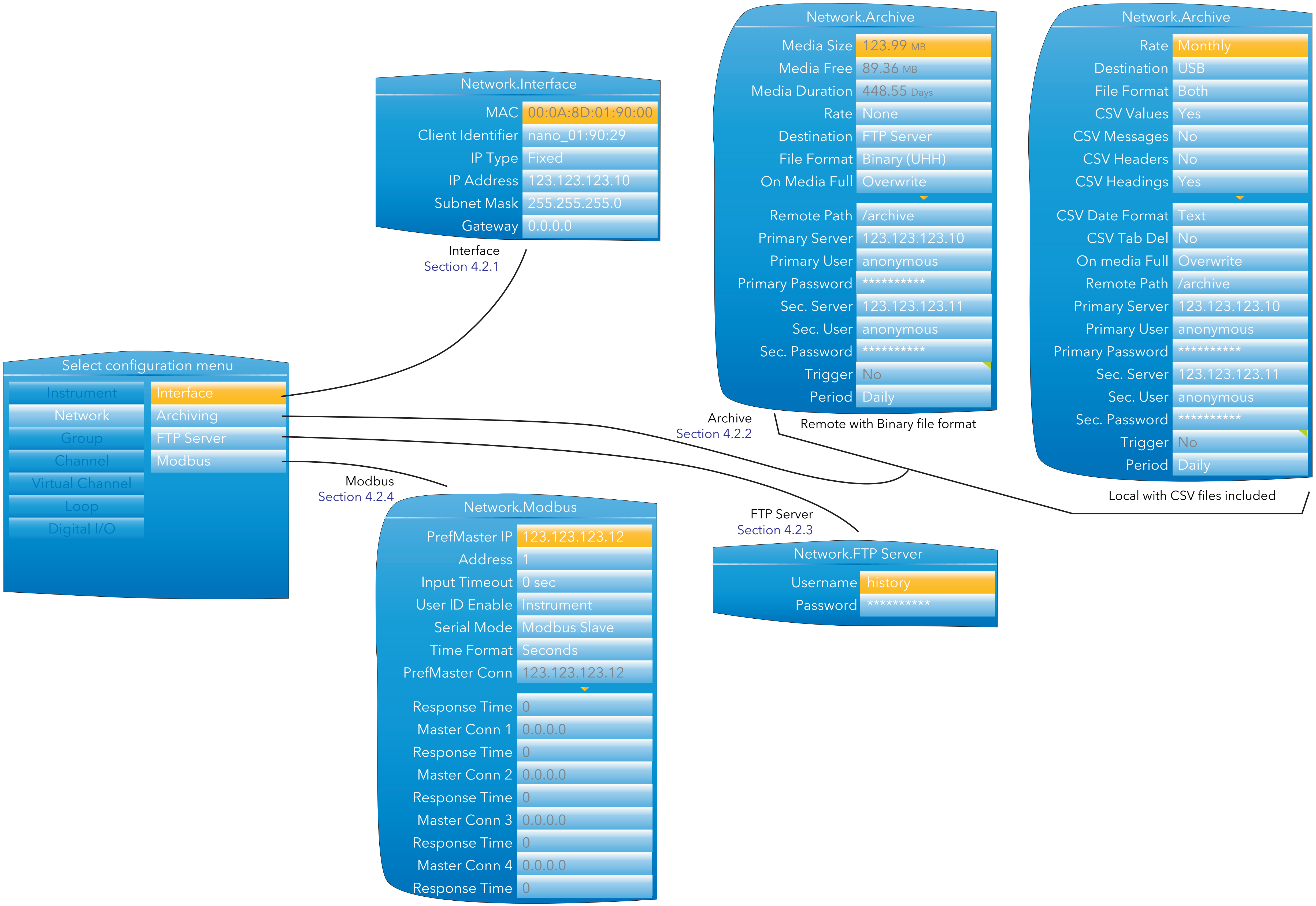



Figure D2 Network configuration menus

## D3 GROUP CONFIGURATION MENU

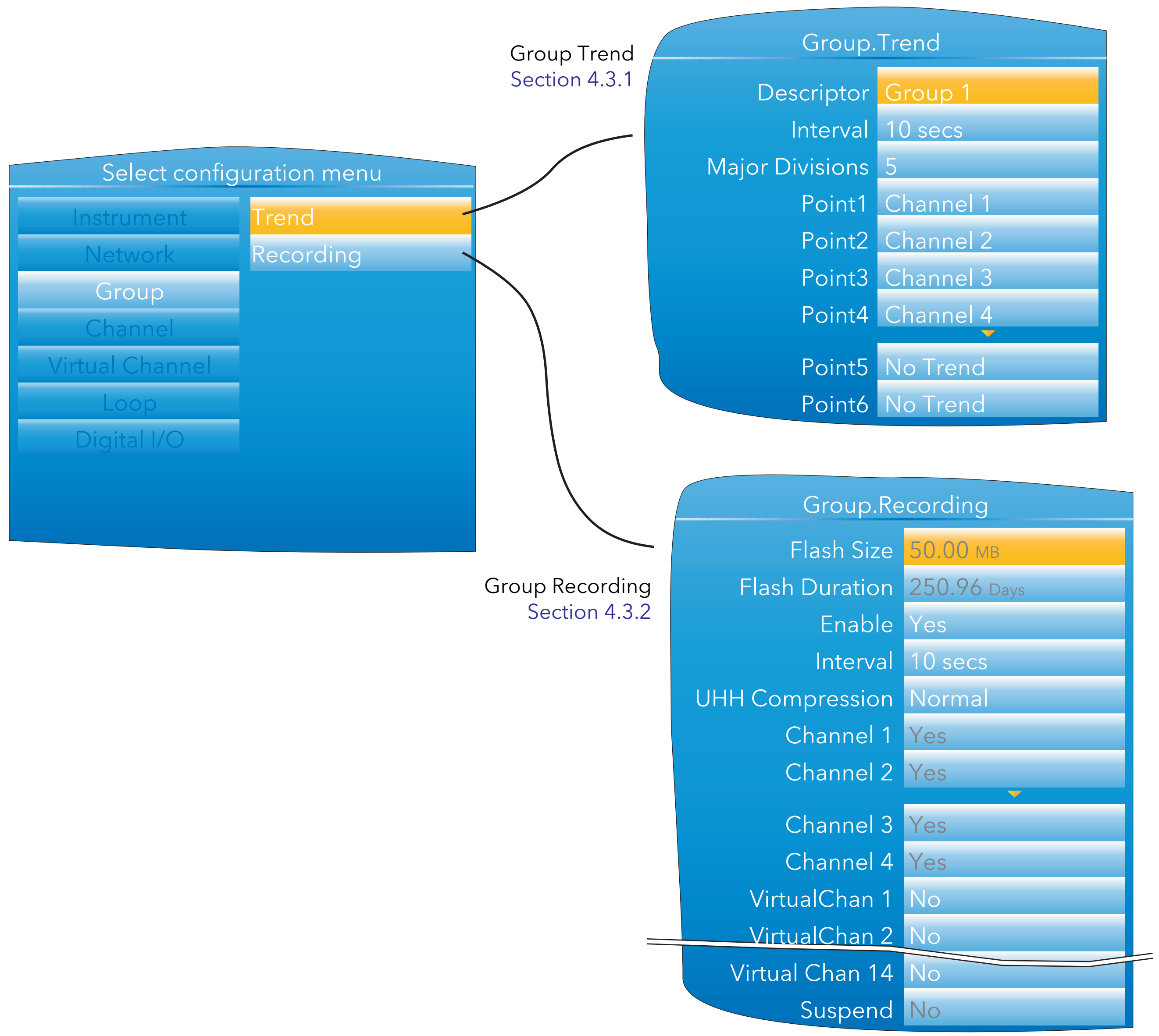



Figure D3 Group configuration menus

## D4 CHANNEL CONFIGURATION MENU

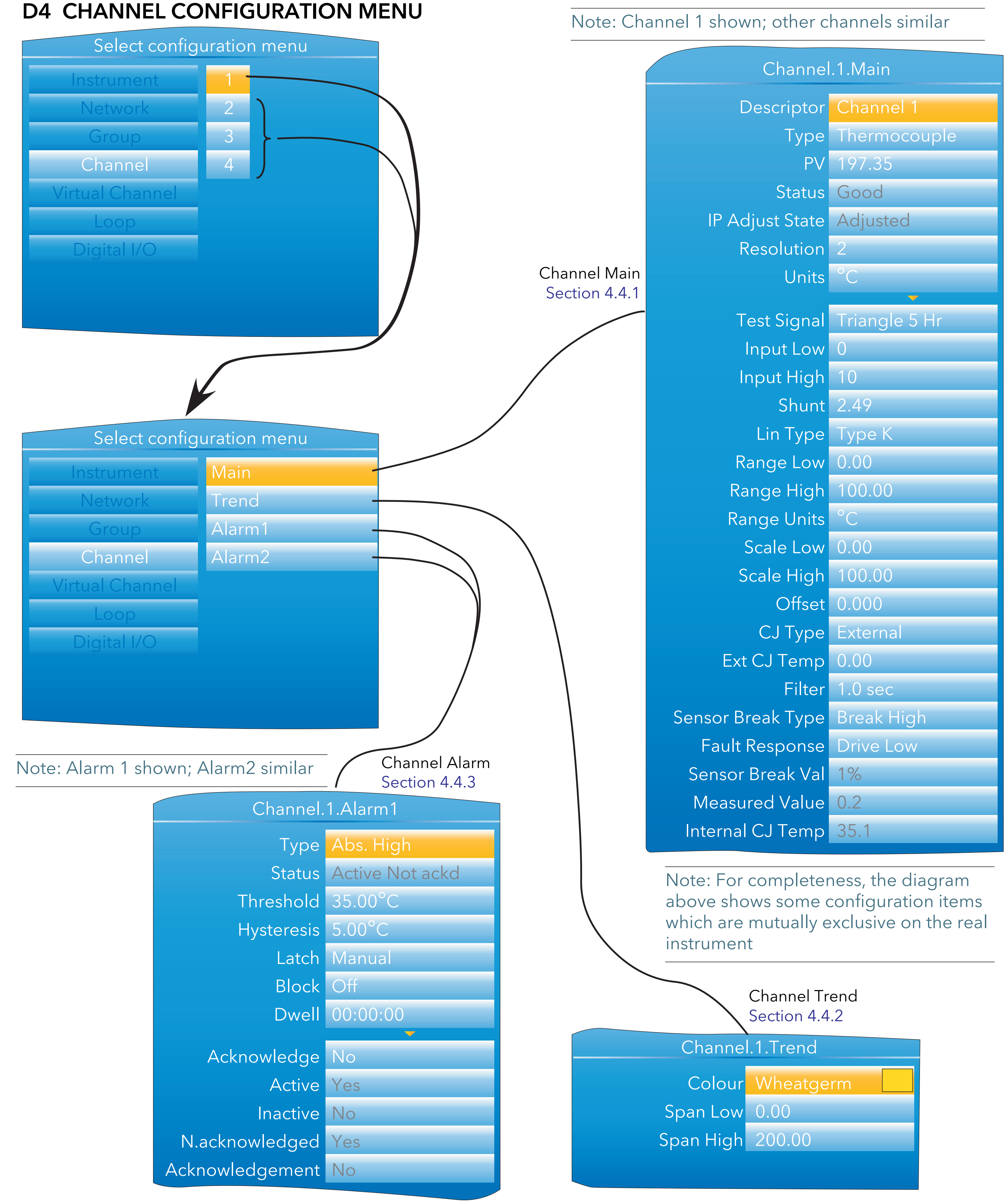
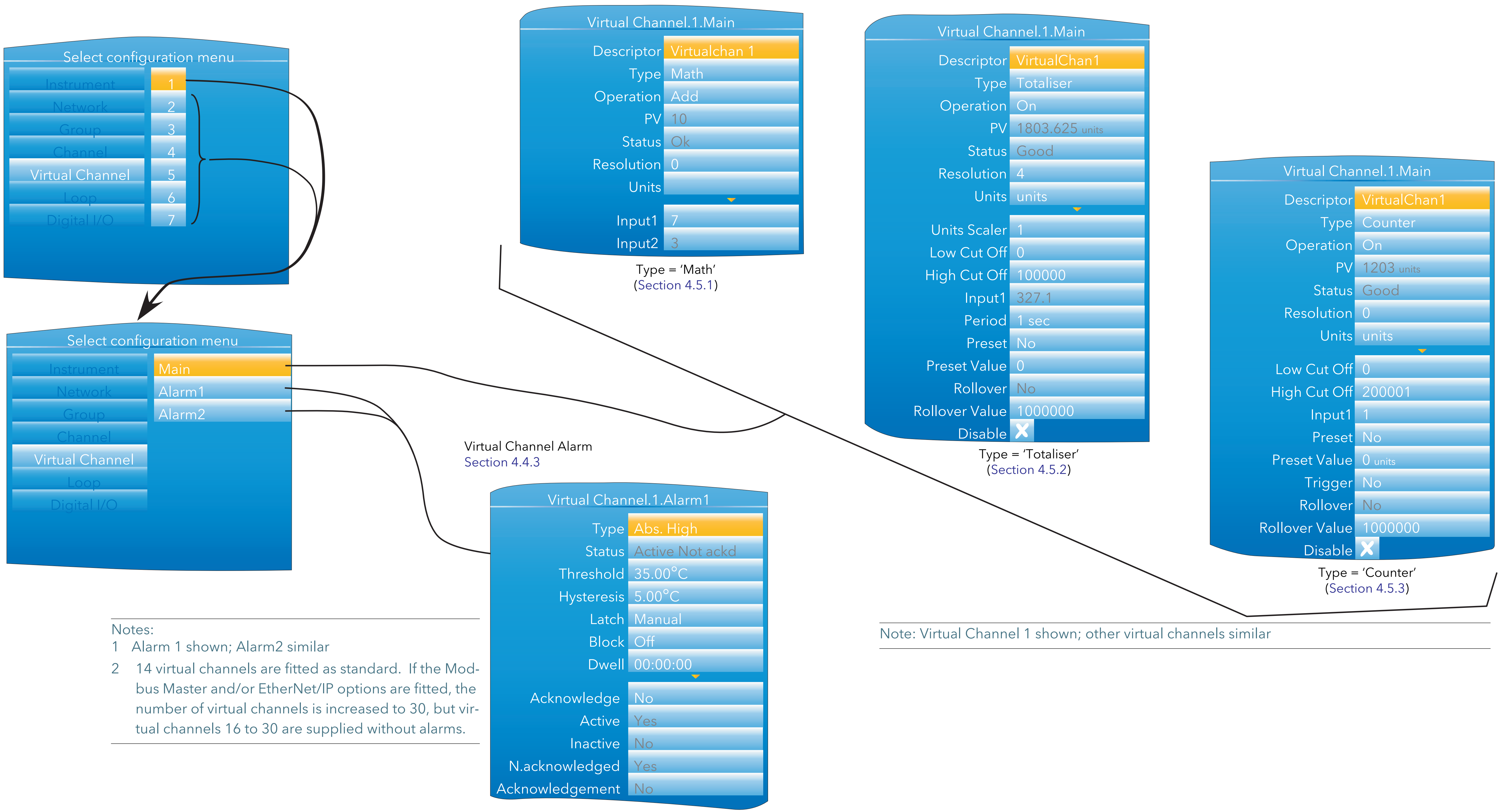



Figure D4 Channel configuration menus

## D5 VIRTUAL CHANNEL CONFIGURATION MENU




Notes:
1 Alarm 1 shown; Alarm2 similar
2 14 virtual channels are fitted as standard. If the Modbus Master and/or EtherNet/IP options are fitted, the number of virtual channels is increased to 30, but virtual channels 16 to 30 are supplied without alarms.

Note: Virtual Channel 1 shown; other virtual channels similar

Figure D5 Virtual channel configuration menus

## D6 LOOP CONFIGURATION MENUS

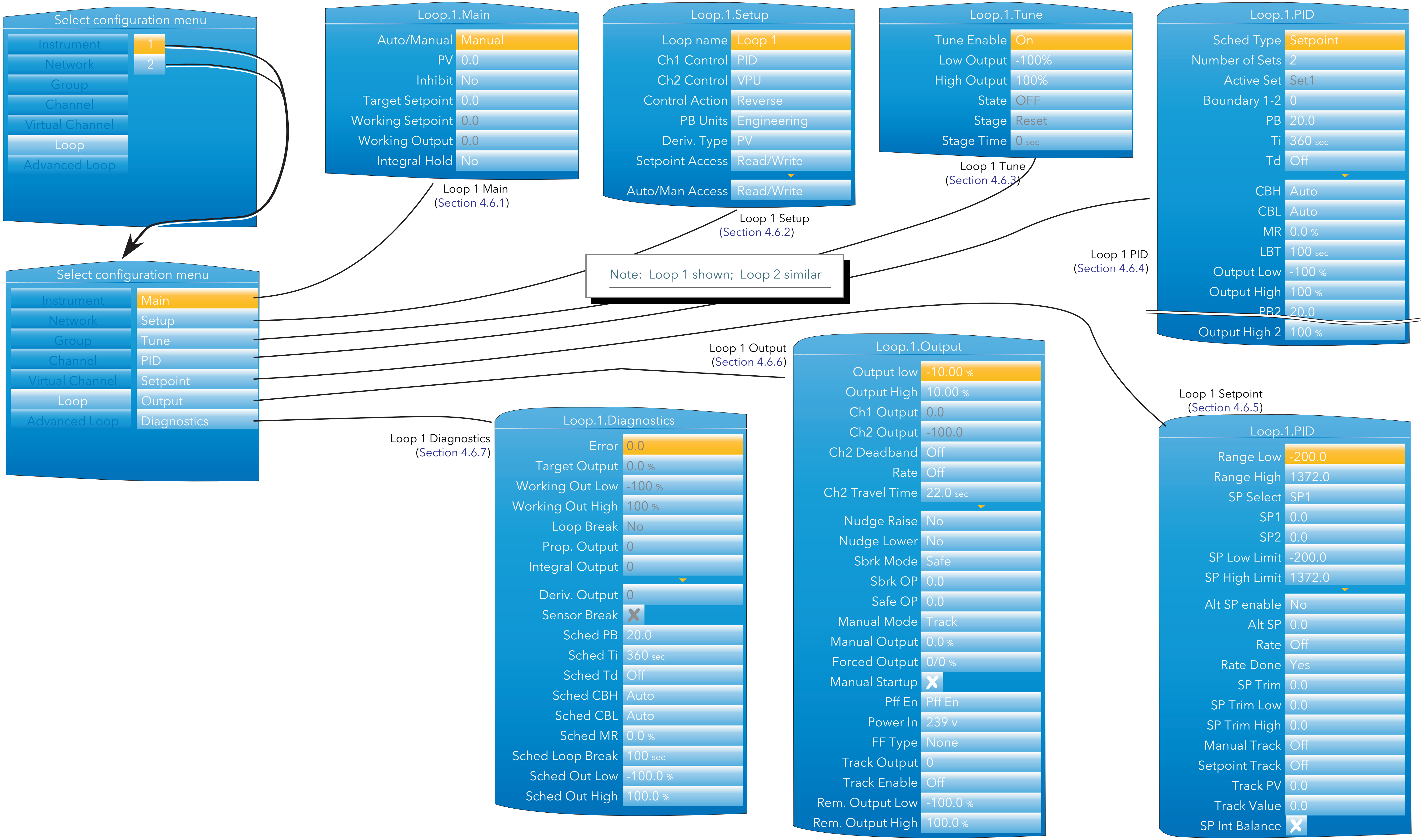



Figure D6 Loop Configuration menus

## D7 ADVANCED LOOP CONFIGURATION MENUS

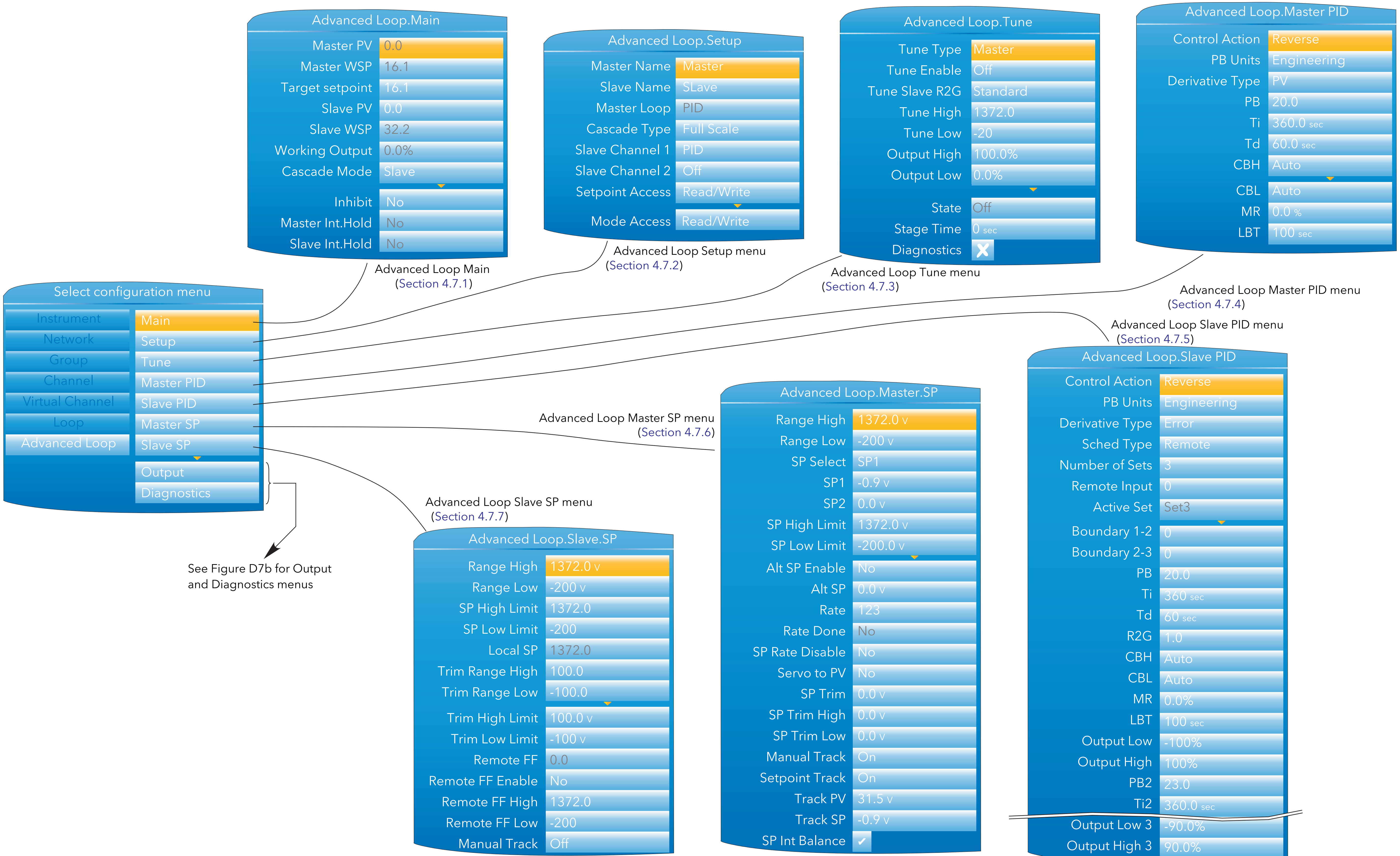



Figure D7a Advanced Loop menus sheet 1

## D7 ADVANCED LOOP CONFIGURATION (Cont.)

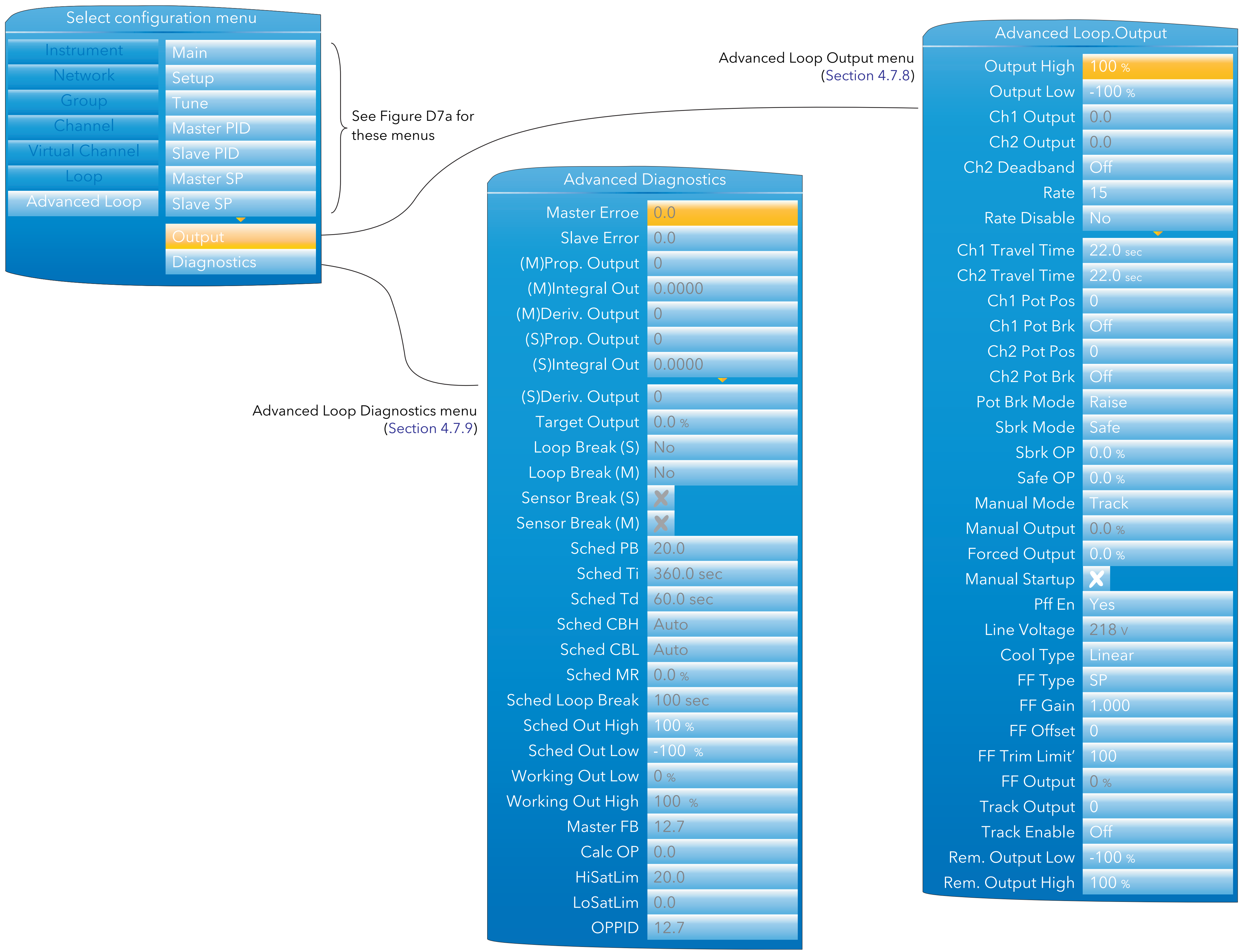



Figure D7b Advanced Loop menus sheet 2

## D8 PROGRAMMER CONFIGURATION

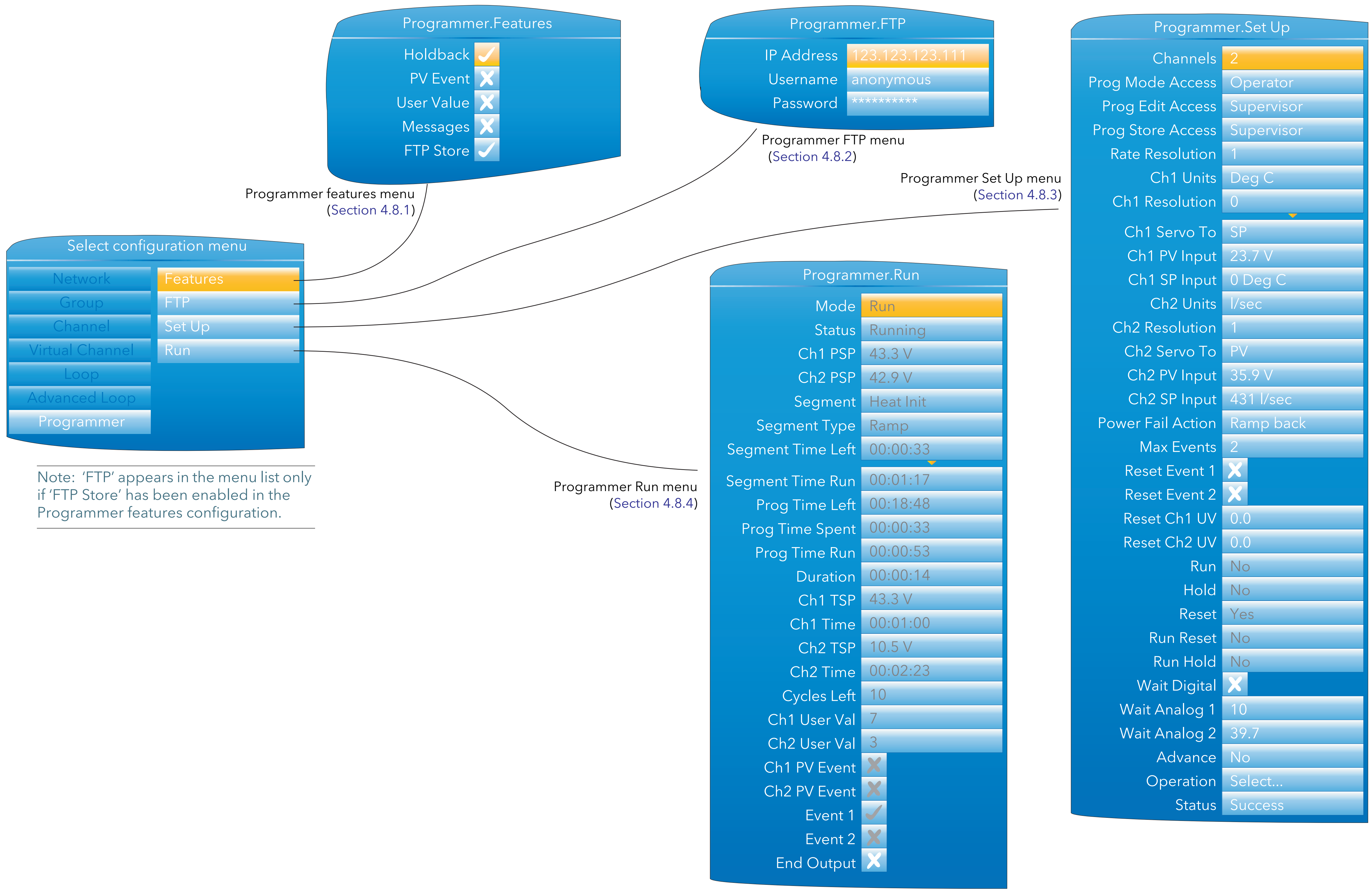



Figure D8 Programmer menus

## D9 MODBUS MASTER CONFIGURATION

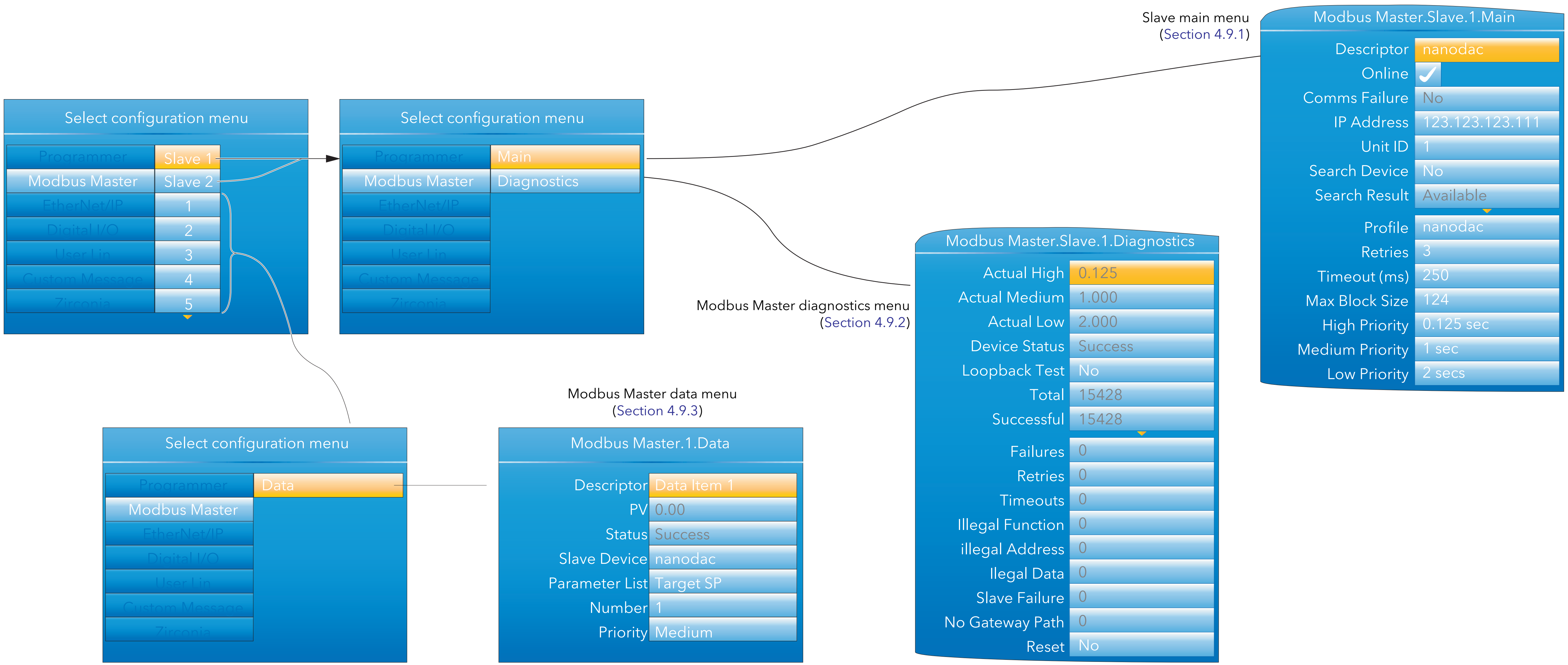



Figure D9 Modbus Master menus

## D10 ETHERNET/IP CONFIGURATION

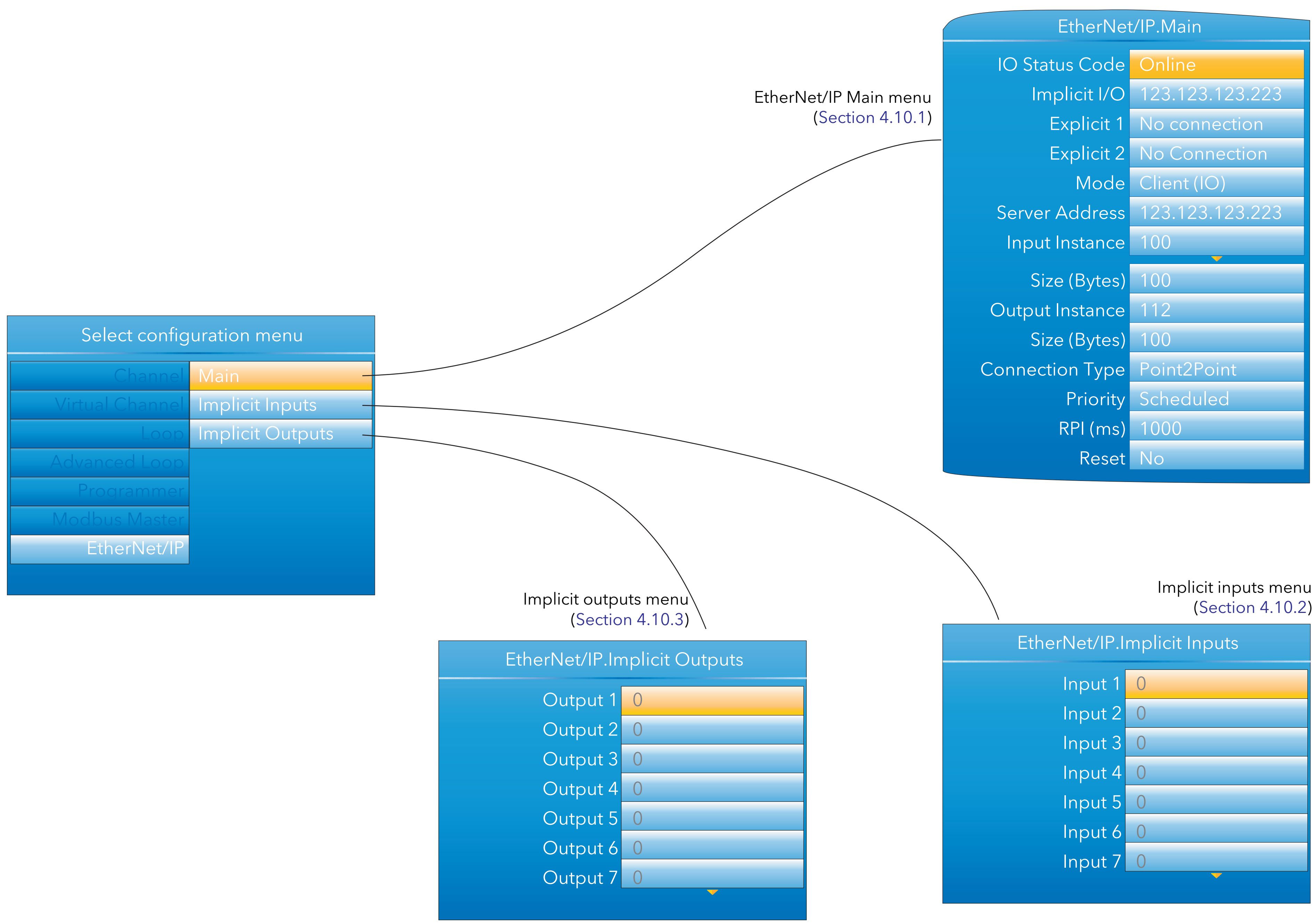



Figure D10 EtherNet/IP menus

## D11 DIGITAL I/O CONFIGURATION MENUS

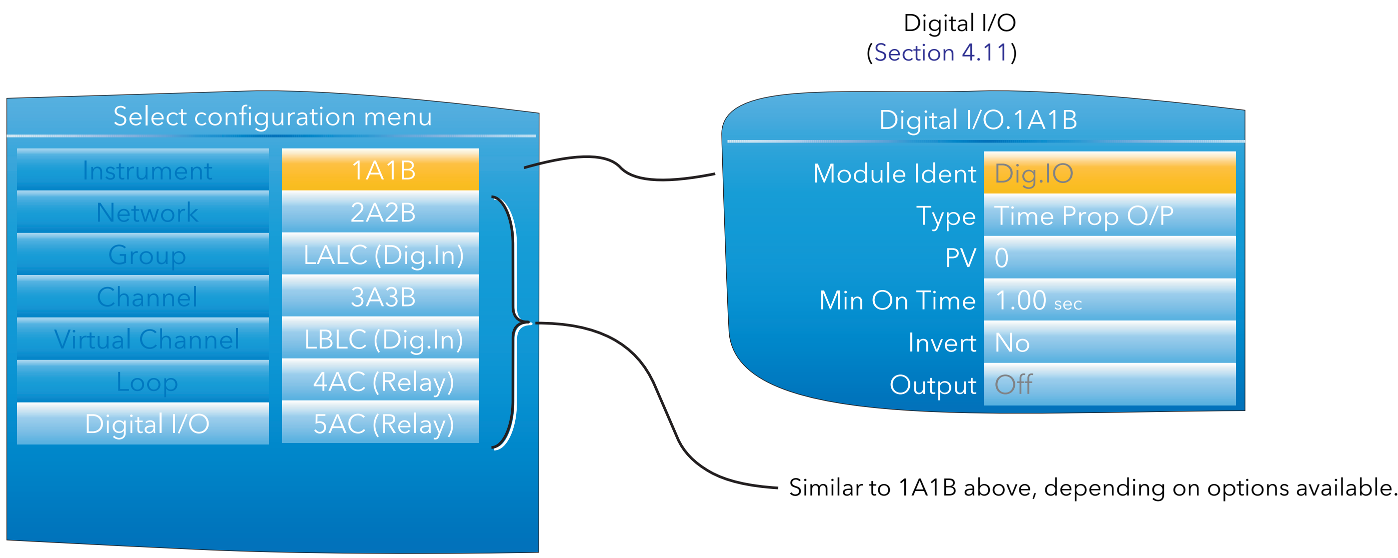



Figure D11 Digital I/O configuration menus

## D12 DC OUTPUT CONFIGURATION MENUS

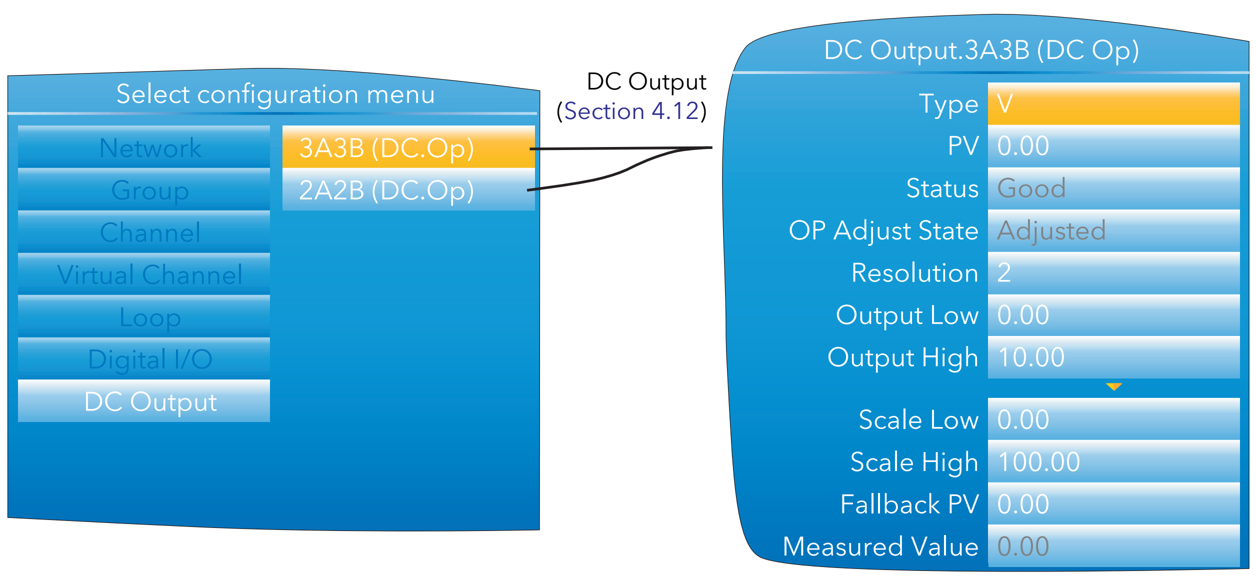



Figure D12 DC output configration menus

## D13 USER LINEARISATION TABLE CONFIGURATION MENU

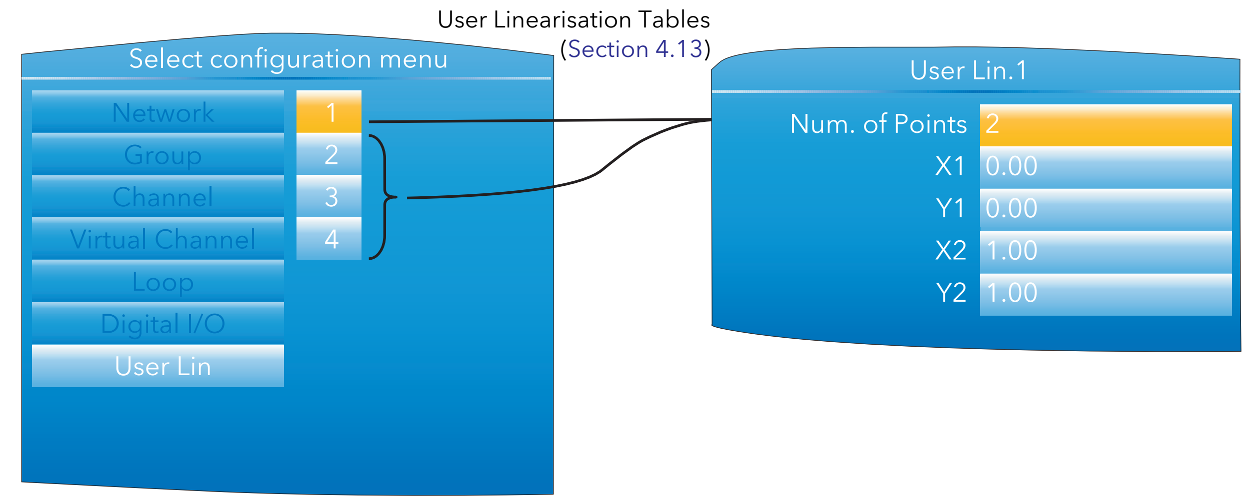



Figure D13 User Linearisation tble menus

## D14 CUSTOM MESSAGES CONFIGURATION MENU

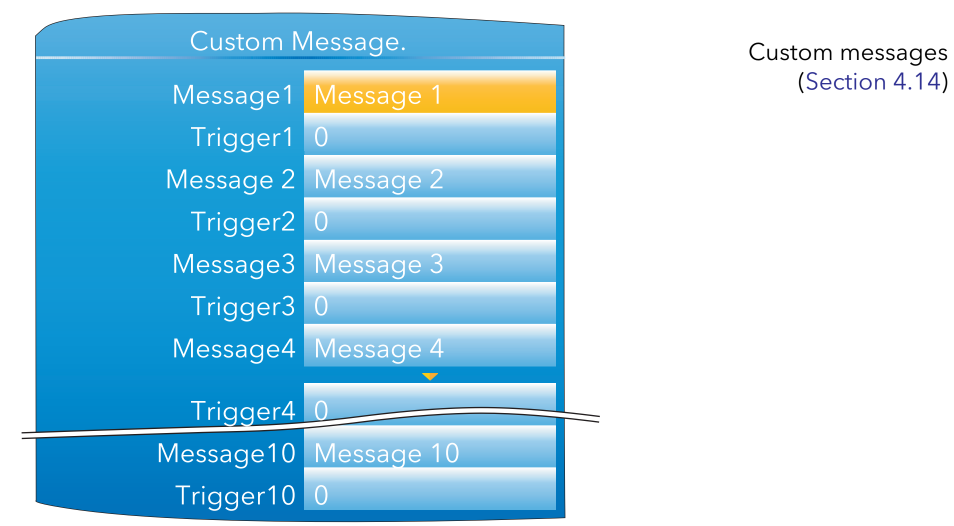



Figure D14 Custom messages configuration

## D15 ZIRCONIA BLOCK CONFIGURATION

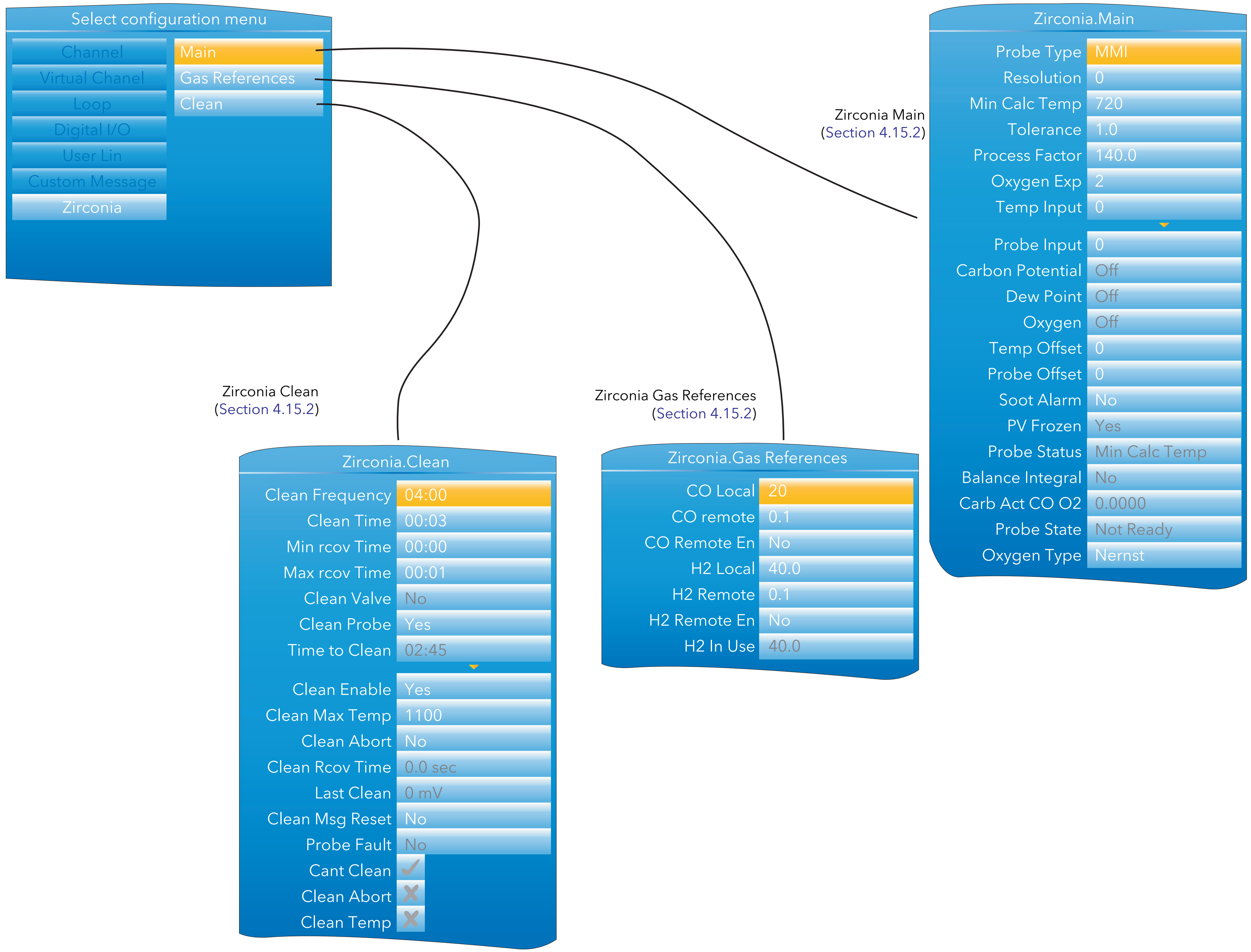



Figure D15 Zirconia block configuration menus

## D16 STERILISER BLOCK CONFIGURATION MENU

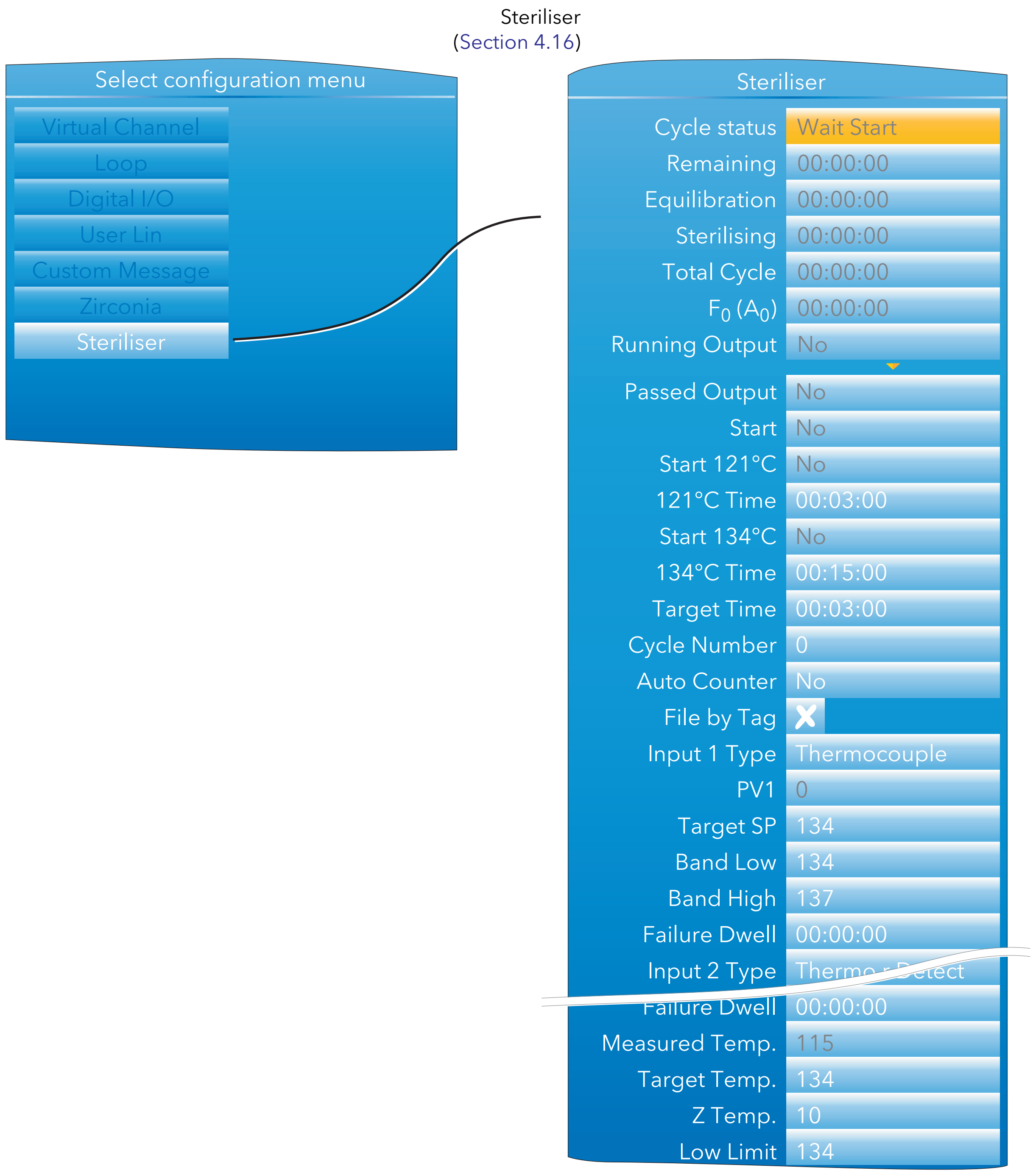



Figure D16 Steriliser menu

## D17 HUMIDITY BLOCK CONFIGURATION MENU

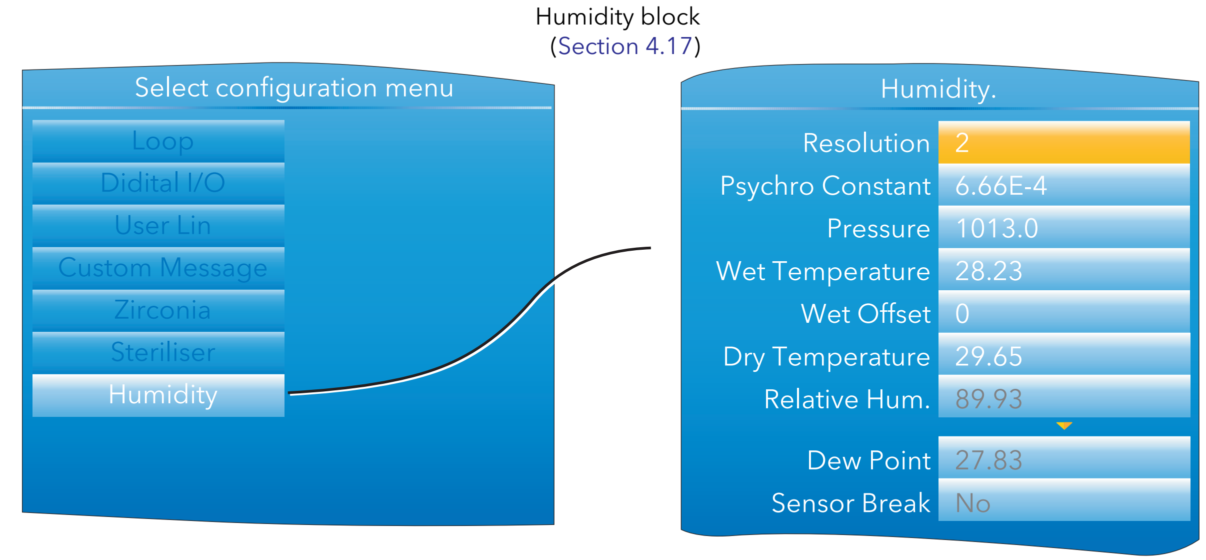



Figure D17 Humidity Block configuration menu

## D18 BCD INPUT BLOCK CONFIGURATION MENU

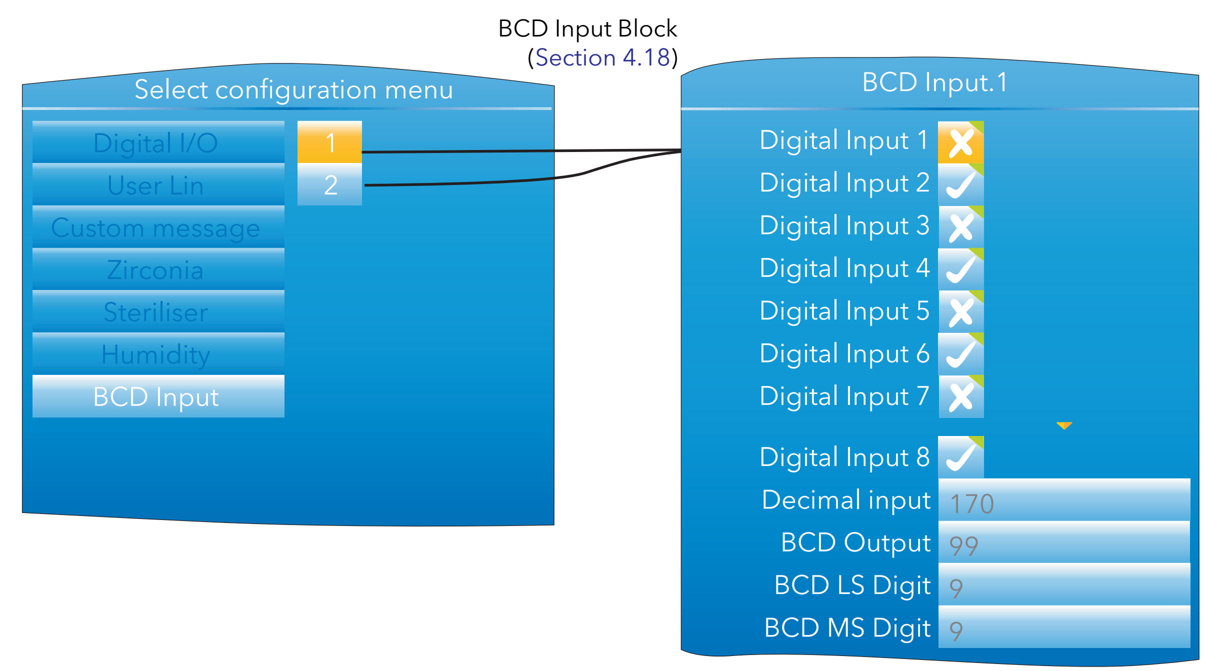



Figure D18 BCD input block menu

## D19 LOGIC (2 INPUT) CONFIGURATION MENU

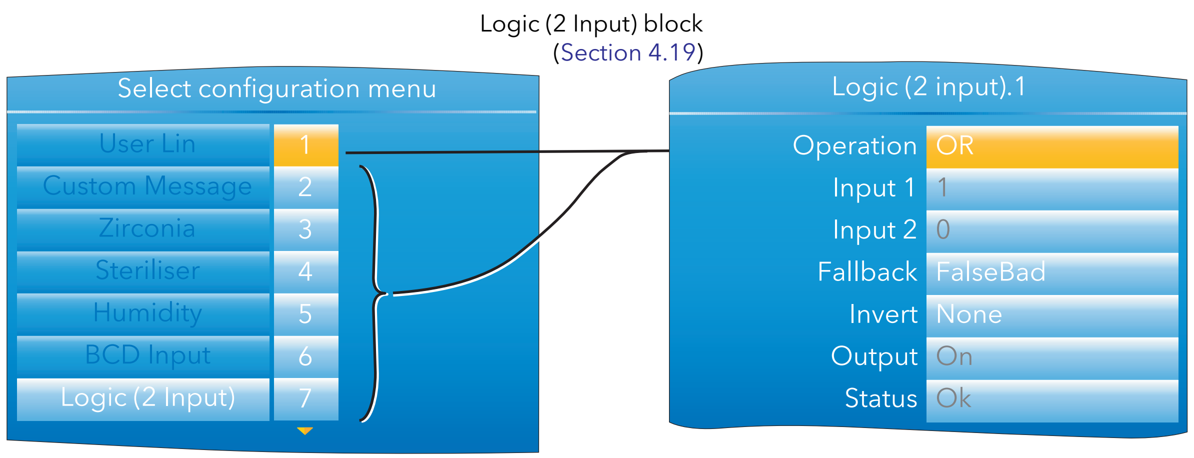



Figure D19 logic (2 input) configuration menu

## D20 LOGIC (8 INPUT) CONFIGURATION MENU

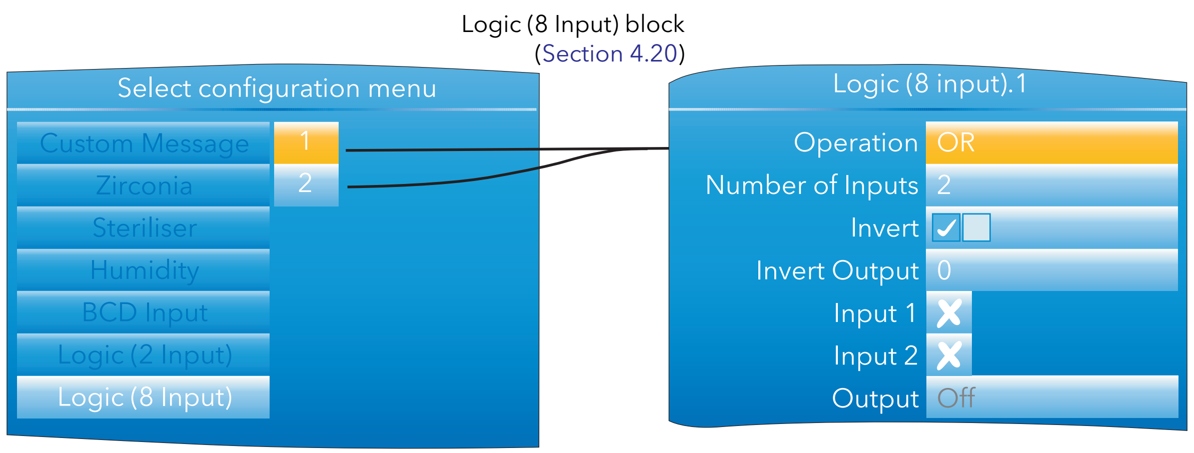



Figure D20 logic (8 input) configuration menu

## D21 MULTIPLEXER BLOCK CONFIGURATION MENU

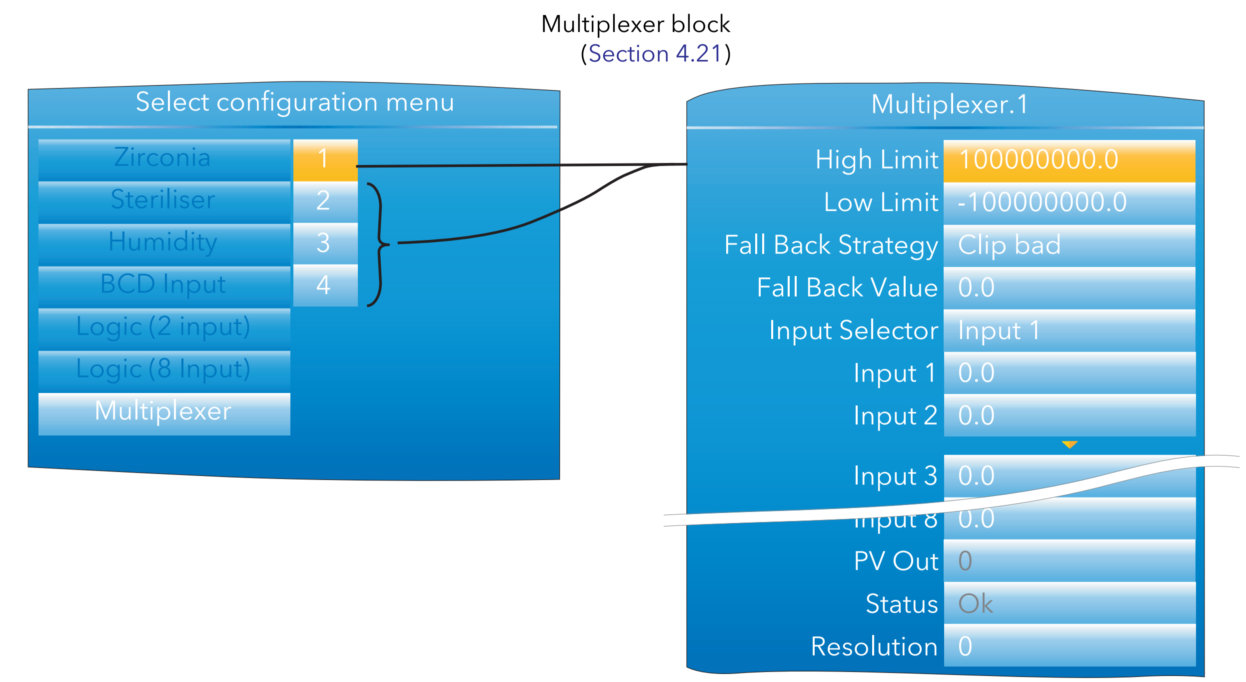



Figure D21 Logic (2 input) configuration menu

## D22 MATH (2 INPUT) CONFIGURATION MENU

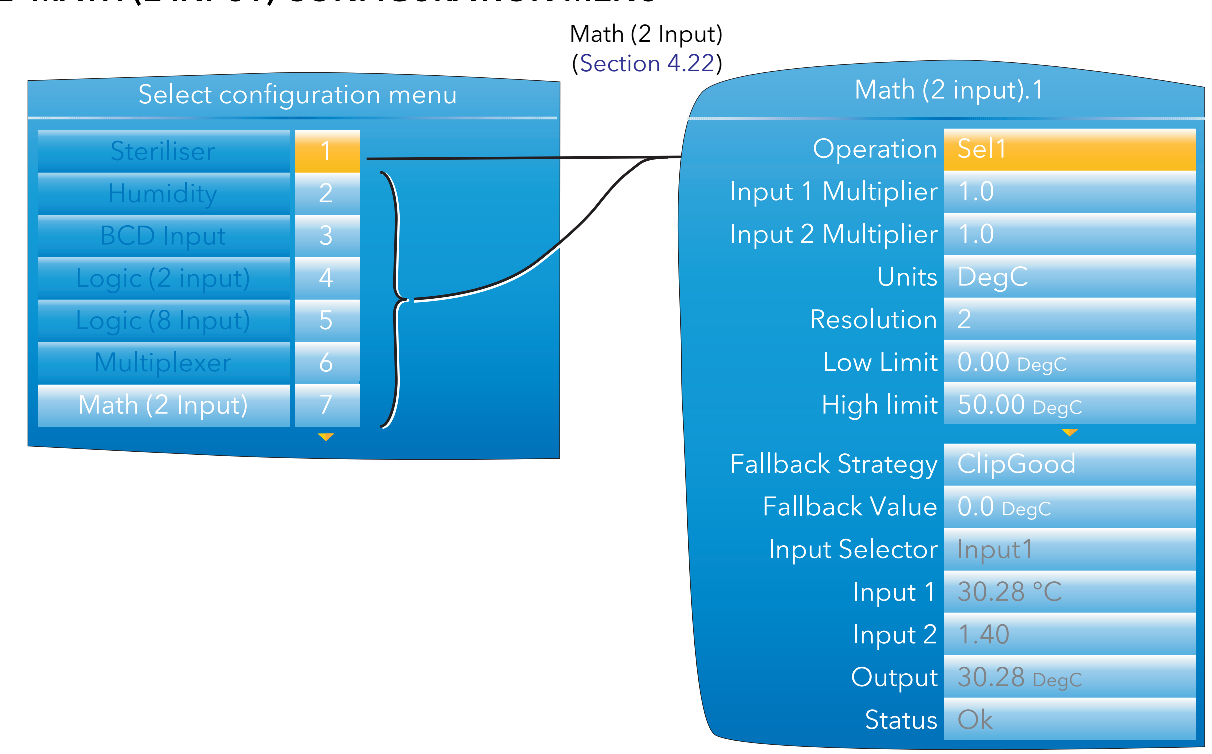



Figure D22 Math (2 Input) configuration menu

## D23 TIMER CONFIGURATION MENU

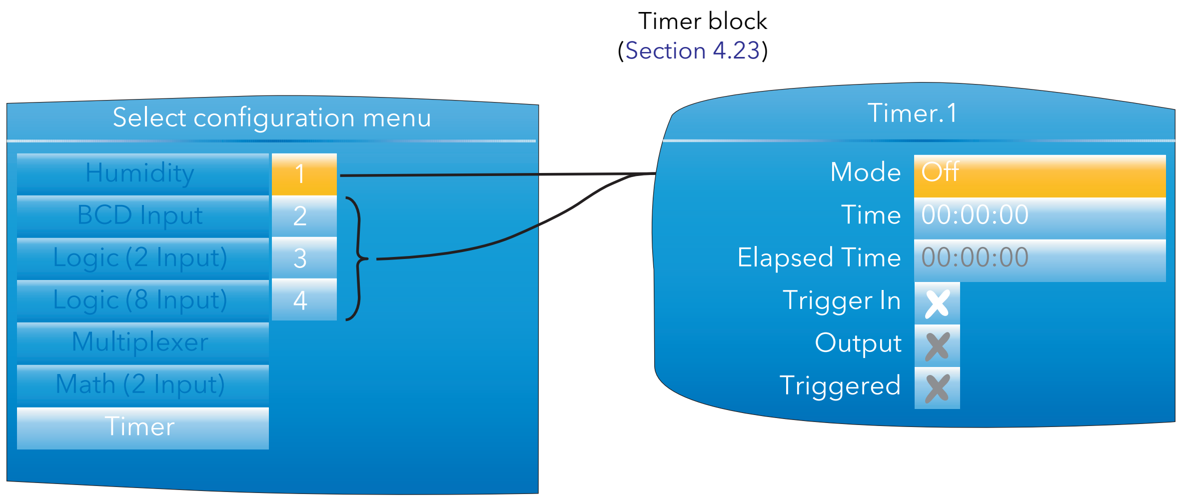



Figure D23 Timer configuration menu

## D24 USER VALUES CONFIGURATION MENU

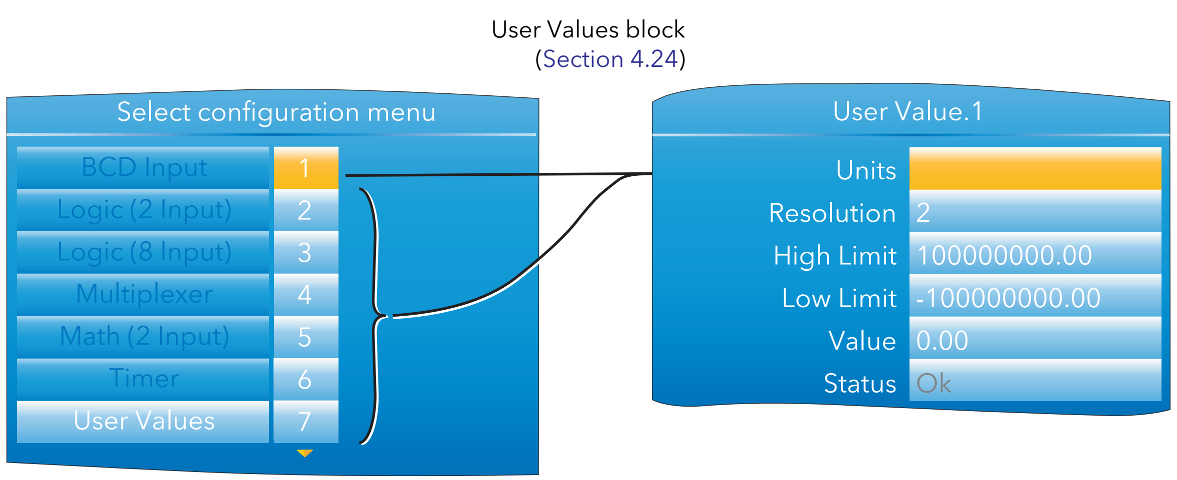



Figure D24 User Values configuration menu

## D25 REAL TIME EVENTS CONFIGURATION MENU

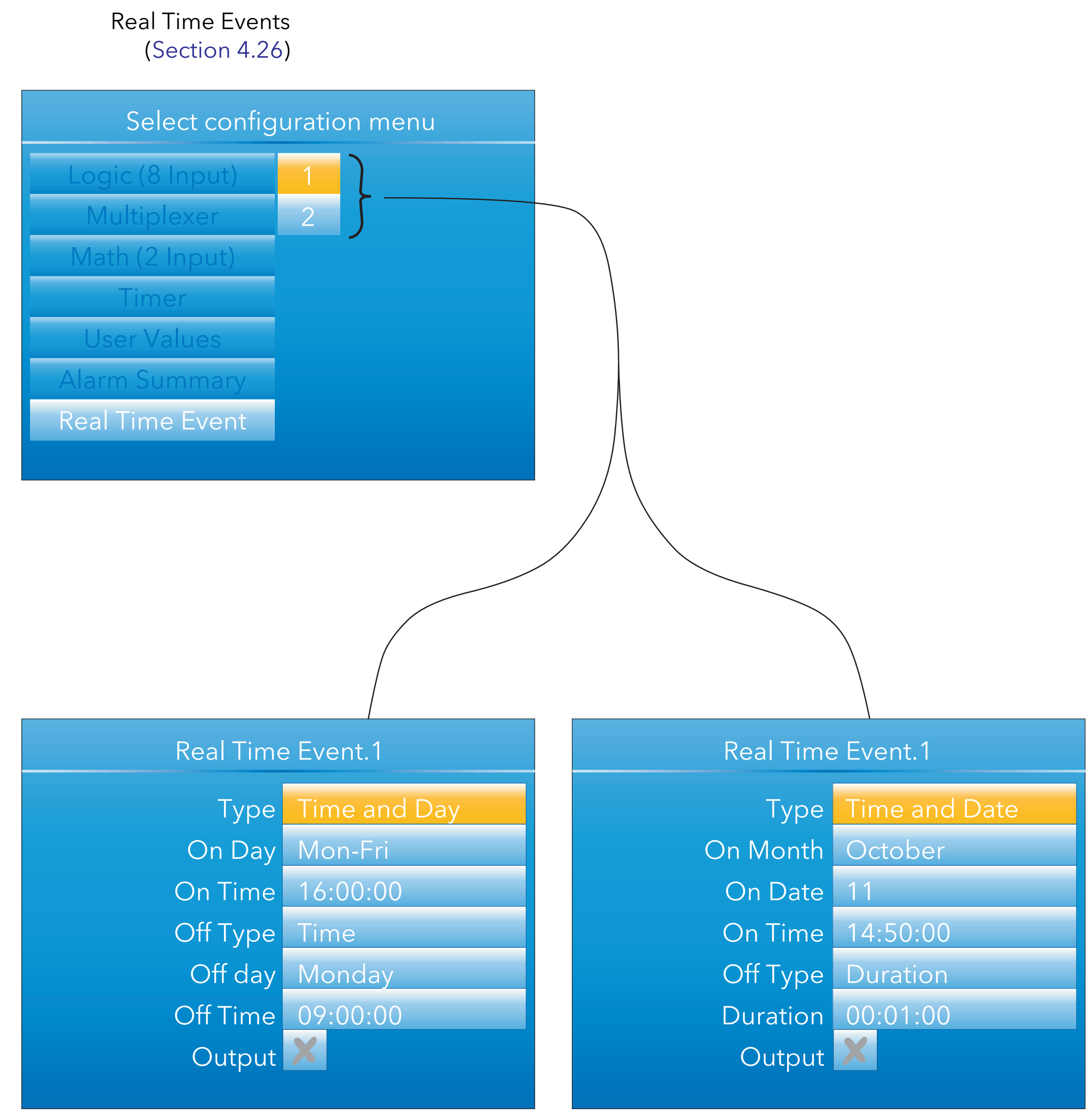



Figure D25 Real Time Event Configration

# Index

## D

## E

## N

## O



## P

## Q



## R

# Eurotherm: International sales and service

**AUSTRALIA Melbourne**
Invensys Process Systems Australia Pty. Ltd.
Telephone (+61 0) 8562 9800
Fax (+61 0) 8562 9801
E-mail info.eurotherm.au@invensys.com

**AUSTRIA Vienna**
Eurotherm GmbH
Telephone (+43 1) 7987601
Fax (+43 1) 7987605
E-mail info.eurotherm.at@invensys.com

**BELGIUM & LUXEMBOURG Moha**
Eurotherm S.A./N.V.
Telephone (+32) 85 274080
Fax (+32 ) 85 274081
E-mail info.eurotherm.be@invensys.com

**BRAZIL Campinas-SP**
Eurotherm Ltda.
Telephone (+5519) 3707 5333
Fax (+5519) 3707 5345
E-mail info.eurotherm.br@invensys.com

**CHINA**
Eurotherm China

**Shanghai Office**
Telephone (+86 21) 61451188
Fax (+86 21) 61452602
E-mail info.eurotherm.cn@invensys.com

**Beijing Office**
Telephone (+86 10) 5909 5700
Fax (+86 10) 5909 5709/10
E-mail info.eurotherm.cn@invensys.com

**FRANCE Lyon**
Eurotherm Automation SA
Telephone (+33 478) 664500
Fax (+33 478) 352490
E-mail info.eurotherm.fr@invensys.com

**GERMANY Limburg**
Invensys Systems GmbH
>EUROTHERM<
Telephone (+49 6431) 2980
Fax (+49 6431) 298119
E-mail info.eurotherm.de@invensys.com

**INDIA Mumbai**
Invensys India Pvt. Ltd.
Telephone (+91 22) 67579800
Fax (+91 22) 67579999
E-mail info.eurotherm.in@invensys.com

**IRELAND Dublin**
Eurotherm Ireland Limited
Telephone (+353 1) 4691800
Fax (+353 1) 4691300
E-mail info.eurotherm.ie@invensys.com

**ITALY Como**
Eurotherm S.r.l
Telephone (+39 031) 975111
Fax (+39 031) 977512
E-mail info.eurotherm.it@invensys.com

**KOREA Seoul**
Invensys Operations Management Korea
Telephone (+82 2) 2090 0900
Fax (+82 2) 2090 0800
E-mail info.eurotherm.kr@invensys.com

**NETHERLANDS Alphen a/d Rijn**
Eurotherm B.V.
Telephone (+31 172) 411752
Fax (+31 172) 417260
E-mail info.eurotherm.nl@invensys.com

**POLAND Katowice**
Invensys Eurotherm Sp z o.o
Telephone (+48 32) 7839500
Fax (+48 32) 7843608/7843609
E-mail info.eurotherm.pl@invensys.com

**POLAND Warsaw**
Invensys Eurotherm Sp z o.o
Telephone (+48 22) 8556010
Fax (+48 22) 8556011
E-mail bieru@invensys-systems.pl

**SPAIN Madrid**
Eurotherm España SA
Telephone (+34 91) 6616001
Fax (+34 91) 6619093
E-mail info.eurotherm.es@invensys.com

**SWEDEN Malmo**
Eurotherm AB
Telephone (+46 40) 384500
Fax (+46 40) 384545
E-mail info.eurotherm.se@invensys.com

**SWITZERLAND Wollerau**
Eurotherm Produkte (Schweiz) AG
Telephone (+41 44) 7871040
Fax (+41 44) 7871044
E-mail info.eurotherm.ch@invensys.com

**UAE DUBAI**
Invensys Middle East FZE
Telephone (+971 4) 8074700
Fax (+971 4) 8074777
E-mail marketing.mena@invensys.com

**UNITED KINGDOM Worthing**
Eurotherm Limited
Telephone (+44 1903) 268500
Fax (+44 1903) 265982
E-mail info.eurotherm.uk@invensys.com

**U.S.A. Ashburn VA**
Invensys Eurotherm
Telephone (+1 703) 724 7300
Fax (+1 703) 724 7301
E-mail info.eurotherm.us@invensys.com

ED68

Contact details correct at time of print.



invensys
Eurotherm

HA030554/6 (CN28666)